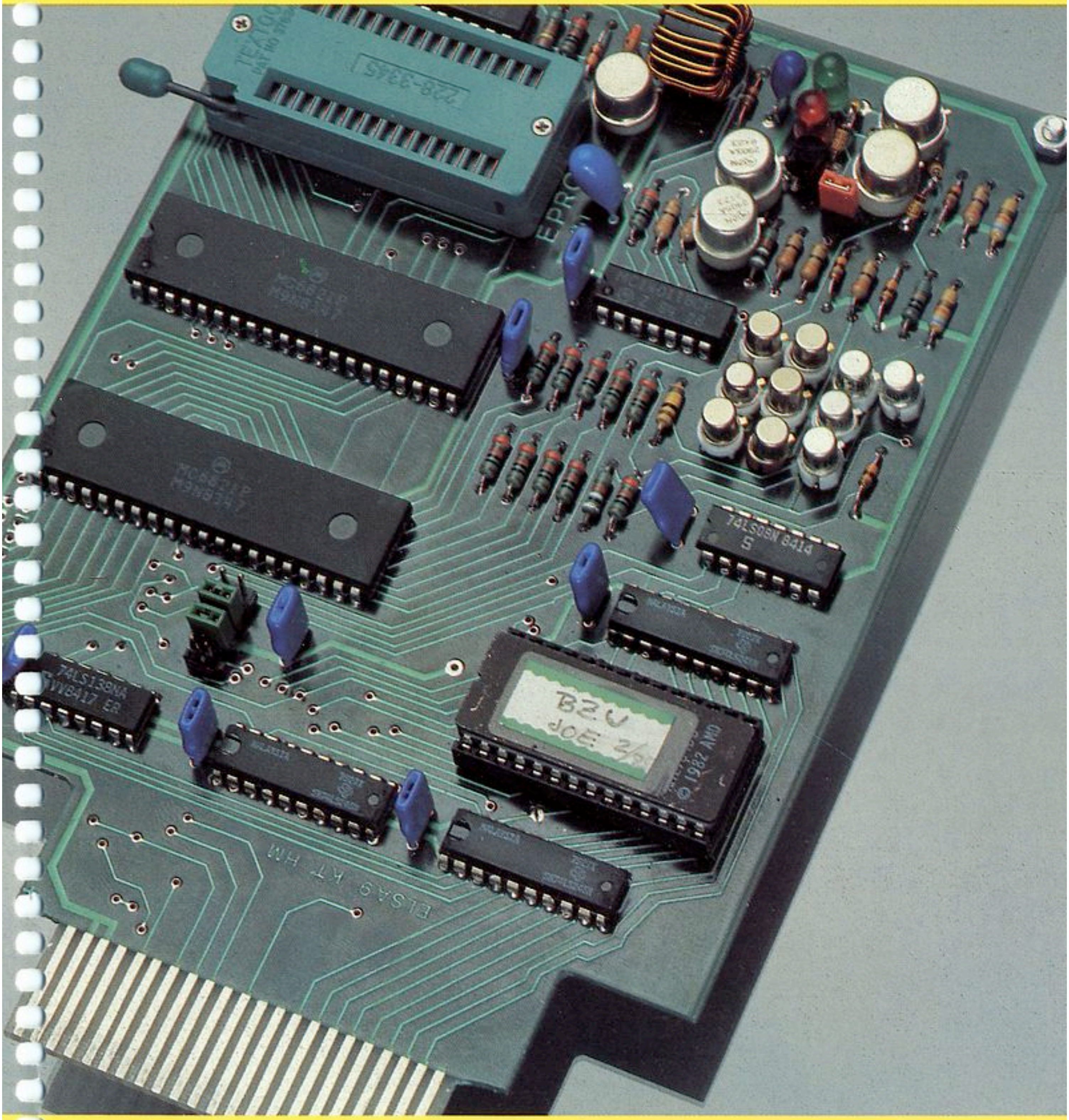
Perusteos elektroniikan harrastajalle.
Rakenna itse AD-muunnin, RS232-liitäntä, virtalähde, rajattoman laaja liitäntä kotimikrolta maailmalle, yleiskäyttöinen digitaalilaskuri, sarjaliikenneohjain.

Bittinikkarin käsikirja



Ari Paananen

BITTI -kirjat

Bittinikkarin käsikirja

Ari Paananen

SANOMAPRINT
ERIKOISLEHDET
TECNOPRESS

Bittinikkarin käsikirja

Ari Paananen

© Sanomaprint, Erikoislehdet, Tecnopress
Kaikki oikeudet pidätetään.

Tämän teoksen tekstiosuuden osittainenkin kopioiminen on tekijänoikeuslain (404/61, muut. 897/80) ja valokuvalain (405/61, muut. 898/80) mukaisesti kielletty ilman kustantajan kirjallista suostumusta lukuun ottamatta tätä teosta koskevia arvosteluja ja katsauksia.

Kustannusosakeyhtiö Otavan painolaitokset, Keuruu, 1992

ISBN 951-832-025-X

Lukijalle

80-luvun alussa alkanut kotitietokoneiden yleistyminen toi vanhaan suosittuun harrastukseen, elektroniikkarakenteluun, aivan uuden ulottuvuuden: kotitietokoneiden lisälaitteiden ja muun digitaalielektroniikan rakentelun.

Mikrojen lisäksi myös elektroniset komponentit ovat jatkuvasti halventuneet ja parantuneet, mikä on heijastunut myös harrastajarakentajien laitteisiin. Alan lehdistössä on tuon tuosta julkaistu toinen toistaan hienompien laitteiden rakennusohjeita.

Tämä kirja käsittelee digitaalisten elektroniikkalaitteiden rakentelua suurempana kokonaisuutena. Kirja lähtee aivan sähkötekniikan alkeista tarjoten hyvän pohjan myös muulle elektroniikkaharrastukselle, mutta keskittyy pian lähes yksinomaan digitaalielektroniikkaan.

Aloittaessani elektroniikka-aiheisten artikkelien kirjoittamisen MikroBITTI-lehteen vuonna 1986 oli erilaisia kotimikroja kaupan todella paljon. Tätä kirjoittaessani markkinat ovat supistuneet pääosin muutamaan laitetyyppiin. Kymmenet tuhannet Commodore 64:t, VIC-20:t ja muut eivät ole kuitenkaan hävinneet minnekään, vaan lojuvat "kakkoskoneina".

Kun itse tehtyjä laitteita kytketään mikroon, on aina vaarana viallisen kytkennän aiheuttama kalliin mikron rikkoutuminen. Niinpä vanha kakkosmikro onkin oiva kohde erilaisten kokeilulaitteiden perustaksi.

Tämän kirjan valmiit sovellukset on pyritty rakentamaan mahdollisimman yleiskäyttöisiksi ja aina jotakin digitaalielektroniikan aluetta opettavaksi. Valitettavasti käytännön laitteita ei kuitenkaan voi rakentaa jokaiseen mikroon sopivaksi. Laitteiden kokeilukappaleista on suuri osa rakennettu C64:ään liitettäväksi, juuri sen yleisyyden vuoksi. Muihin mikroihin liitettäessä laitteen rakentaja joutuu tekemään muutoksia yleensä sekä laitteeseen että sen ohjausohjelmistoon.

Kirjan ensimmäiset kahdeksan kappaletta käsittelevät elektroniikan teoriaa. Suosittelen, että lukisit nämä kappaleet järjestyksessä, koska asiat niissä pohjautuvat vahvasti aiempiin kappaleisiin. Loppuosa kirjasta koostuu itsenäisistä sovelluksista, joihin voi mainiosti tutustua haluamassaan järjestyksessä.

Toivon vilpittömästi, että saat Bittinikkarin käsikirjaa lukiessasi siitä runsaasti iloa, tietoa ja virikkeitä harrastukseesi.

Vantaalla 15.1.1992

Ari Paananen

Sisältö

1. Sähköopin perusteet 7

Sähkövirta 8
Jännite 8
Vastus 9
Teho 9
Ohmin laki 10
Kirchhoffin lait 11
Kapasitanssi 12
Induktanssi 13
Vaihtosähkö 14
Suureiden etuliitteet 14

2. Käytännön passiiviset komponentit ...16

Vastukset 16
Vastusarvon toleranssi 16
Vastuksen tehonkesto 17
Vastusten värikoodaus 17
Kondensaattorit 19
Kuivat kondensaattorit 20
Elektrolyyttikondensaattorit 20
Säätökondensaattorit 21
Kondensaattorin jännitteenkesto 21
Kelat ja muuntajat 21

3. Puolijohteet 23

Puolijohteiden teoriaa 23
Diodi 24
Transistori 25

4. Käytännön puolijohdekomponentit 28

Diodi 28
Transistorit 30
Mikropiirit 33

5. Digitaalitekniikka 34

Binäärikoodi 34
Perusportit 37
Porttien sovelluksia 42
Kiikut (flip-flop) 44
RS-kiikku 45
D-kiikku 47
JK-kiikku 47
Kiikkujen liipaisutavat 48
Kiikkujen sovelluksia 49

6. Käytännön digitaalipiirit 51

TTL-perhe 51
MOS-perheet 52
Muita logiikkaperheitä 53
Eri logiikkaperheiden erot 54

7. Suuret mikropiirit 55

Muistit 55
Muut erikoispiirit 61

8. Mikrotietokoneen perusrakenne 62

Väylät 62
Mikrotietokoneen väylät 62
Prosessori ja konekieli 67

Virtalähteet ja jännitteen vakavointi 73

A/D-muuntimen sovelluksia 77

Näytteenottotaajuus 78
Laitteen liitäntä mikroon 78
A/D-muuntimen rakenne 79
D/A-muunnin 79
Muunninpiirin toiminta 80

Muunninpiirin kontrollisignaalit 80
Rakentaminen 81
Toiminnan testaus 84
Käyttö C64:n kanssa 85
Liitäntä PC-mikroon 86
C64-oskilloskooppi 91
Suodatin 92

Rajattoman laaja liitäntä mikrolta ulkomaailmaan 119

Ohjausohjelman runko 123
Ketjun pidennys 123

Yleiskäyttöinen digitaalilaskuri 126

Sarjaliikenneohjain 135

Sanasto 139

Indeksi 150

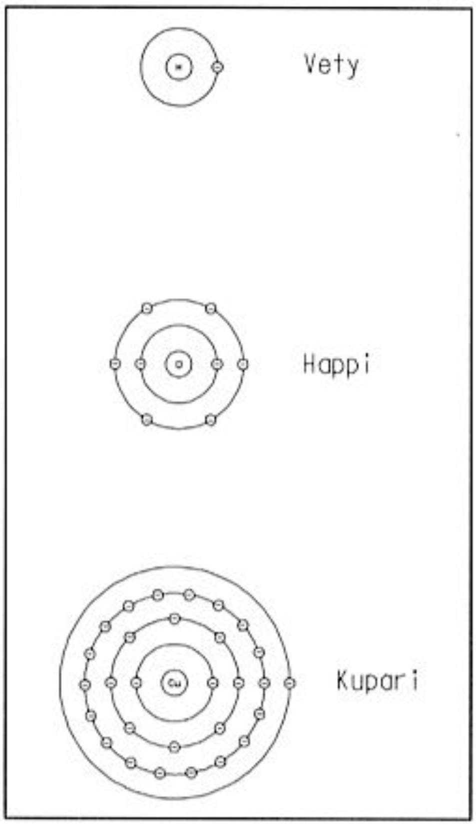
1. Sähköopin perusteet

Ymmärtääkseen sähköoppia, pitää ensin tuntea hieman aineiden rakenteita. Alkuaineet koostuvat pienistä hiukkasista, joita kutsutaan atomeiksi. Atomi on siinä mielessä jakamaton, että jos se hajoitetaan, muuttuu ainekin toiseksi. Atomi koostuu kolmentyyppisistä hiukkasista, ns. alkeishiukkasista. Alkeishiukkaset ovat suuruusjärjestyksessä lueteltuina neutroni, protoni ja elektroni. Protonit ovat ovat positiivisesti ja elektronit negatiivisesti varautuneita hiukkasia kun taas neutronit ovat varauksettomia.

Neutronit ja protonit muodostavat atomissa ytimen. Atomiydintä kiertää yksi tai useampi elektroni. Kuvassa 1.1 on esitetty muutaman alkuaineen atomien kaavakuvat.

Kaikista alkuaineista yksinkertaisin on vety, jonka ytimen muodostaa yksi protoni, jota puolestaan kiertää vain yksi elektroni. Toinen esimerkkiaine on happi ja kolmas kupari. Elektronit kiertävät ydintä tiettyjä ratoja pitkin. Jokaisella ratatasolla, ns. elektronikuorella, voi olla korkeintaan tietty määrä elektroneja. Sisimmällä radalla tämä määrä on 2, toisella 8, kolmannella 18 jne. Uloimmalla radalla ei kuitenkaan koskaan voi olla yli kahdeksaa elektronia.

Jos aineen uloimmalla kuorella on kuparin tapaan vain yksi elektroni, voi tuo uloin irrallinen elektroni helposti irrota atomista ja lähteä vaeltamaan muiden atomien kuorille. Tällaisia aineita sanotaan johteiksi, koska ne johtavat hyvin sähköä juuri irrallisten elektroniensa ansiosta. Useimpia muita aineita sanotaan puolestaan eristeiksi. Johteita ovat esimerkiksi useimmat metallit, ja eristeitä esimerkiksi lasi ja puhdas vesi. Lisäksi on olemassa ns. puolijohteita, jotka ovatkin mikroilijoiden kannalta suorastaan elintärkeitä aineita, mutta niitä käsitellään tarkemmin myöhemmin.



Kuva 1.1. Muutaman alkuaineen atomien kaavakuvat.

Sähkövirta

Kun elektronit liikkuvat johteessa, sanotaan että siinä kulkee sähkövirta. Sähkövirta johtimessa vastaa hyvin veden virtausta joessa. Kuvitellaanpa jokea, jonka yli kulkee silta. Jos sillan ali yhden sekunnin aikana virtaavan veden määrä mitattaisiin, saataisiin veden virran voimakkuutta kuvaava luku. Virran voimakkuus voisi olla esimerkiksi 1 531 967 litraa sekunnissa.

Aivan samoin sähkövirtaa mitataan joen uomaa vastaavan johtimen läpi kulkevien elektronien lukumäärän mukaan. Käytännön sähkövirrat ovat niin suuria, että yksikön "elektronia sekunnissa" sijasta on käyttöön otettu tämän monikerta. Sähkövirran suuruus ilmaistaan yksiköllä ampeeri. Jos johtimen poikkipinnan läpi kulkee sekunnin aikana noin 6,3 miljardia miljardia elektronia, kulkee siinä silloin suuruudeltaan yhden ampeerin virta.

Ampeerin lyhenne on A-kirjain ja matemaattisissa kaavoissa sähkövirtaa merkitään I-kirjaimella.

Jännite

Toinen välttämätön peruskäsite on jännite. Jos sähkövirtaa verrataan joen poikkileikkauksen läpi tietyssä ajassa virtaavan veden määrään, niin jännitettä puolestaan vastaa joen ylä- ja alajuoksun välinen korkeusero. Jos korkeuseroa ei ole, ei myöskään vesi virtaa. Aivan samoin sähkövirran edellytyksenä on kahden pisteen välisen jännitteen olemassaolo.

Jännitteen yksikkö on voltti, lyhennettynä V. Kaavoissa jännitteen tunnus on virallisesti U-kirjain. Joskus käytetään hämäävästi V:tä sekä jännitteen että voltin lyhenteenä.

Jännitteellä on suuruuden lisäksi olemassa napaisuus. Jo sähkötutkimusten alussa huomattiin, ettei ollut sama miten päin virtalähde kytkettiin. Niinpä sovittiin, että toista napaa kutsutaan positiiviseksi tai plusnavaksi ja toista negatiiviseksi tai miinusnavaksi.

Samalla sovittiin, että virta kulkee positiivisesta navasta negatiiviseen. Pahaksi onneksi myöhemmin kuitenkin huomattiin, että elektronit, jotka oikeastaan toimivat varauksen kuljettajina, kulkevatkin juuri päinvastaiseen suuntaan. Niinpä vielä nykyäänkin sekaannuksia aiheuttaa se, että sähkövirta ja elektronit kulkevat eri suuntiin.

Vastus

Tarvitsemme vielä yhden käsitteen tai hienosti sanottuna suureen ennen kuin voimme koota ensimmäisen matemaattisen yhtälön. Tämä suure on nimeltään vastus tai resistanssi. Johtimen resistanssi kuvaa sitä vastustusta, jonka johdin aiheuttaa sähkövirralle. Kun johtimen vastus on suuri, on sähkövirran kulku vaivalloisempaa kuin pienellä vastuksella.

Sähköistä vastusta jokiympäristössä vastaa esimerkiksi kivien ja joen uoman muodostama veden kulkua hidastava liikevastus. Vastuksen symboli kaavoissa on R-kirjain ja yksikkönä käytetään ohmia, jonka lyhenne on kreikkalaisten aakkosten iso omega-kirjain. Sähköistä komponenttia, jolla saadaan aikaan halutun suuruinen vastus, kutsutaan resistoriksi tai tutummin vastukseksi.

Teho

Käytännön kytkennöissä joudutaan tekemisiin vielä yhden uuden suureen, tehon, kanssa. Yleisesti teho ilmaisee kuinka paljon energiaa kuluu, siirtyy tai tuotetaan tietyssä ajassa. Tehomerkinnät ovat useimmille tuttuja esimerkiksi kodinkoneista ja autoista. Tehon symboli on P ja yksikkö on watti eli W.

Esimerkiksi tavallisen sähkölieden keittolevyn teho on 1 000 wattia, joten päällä ollessaan se kuluttaa sähköenergiaa 1 000 watin teholla. Loviisan ydinvoimalaitos puolestaan tuottaa sähköenergiaa 900 000 000 watin teholla, joten se pystyisi pitämään lähes miljoona keittolevyä kuumana.

TEHO = JÄNNITE x VIRTA	(P = U x I)
TEHO = VASTUS x VIRTA x VIRTA	(P = R x I x I)
TEHO = JÄNNITE x JÄNNITE / VASTUS	(P = U x U / R)

Taulukko 1.1. Tehon, jännitteen, virran ja resistanssin riippuvuus.

JÄNNITE = VIRTA x VASTUS	(U = I x R)
VIRTA = JÄNNITE / VASTUS	(I = U / R)
VASTUS = JÄNNITE / VIRTA	(R = U / I)

Taulukko 1.2. Ohmin laki kolmella kaavalla esitettynä.

Jos tuntemattoman vastuksen päiden välillä on 12 V jännite ja sen läpi kulkee 0,05 A virta, voidaan vastuksen kuluttama teho laskea taulukon 1.1 ensimmäisellä kaavalla:

```
P = U x I = 12 V x 0,05 A = 0,6 W
```

Jos taas 100 ohmin vastuksen läpi kulkee 0,03 A virta, voidaan sen kuluttama teho laskea taulukon 1.1 keskimmäisellä kaavalla:

```
P = R x I x I = 100 ohmia x 0,03 A x 0,03 A = 0,09 W
```

Ohmin laki

Jännite, virta ja vastus ovat toisistaan riippuvaisia taulukossa 1.2 esitetyn ns. Ohmin lain mukaisesti. Huomaa, että kaikki taulukon kolme kaavaa ovat vain yhden ja saman kaavan eri näkökulmista katsoen tehtyjä versioita.

Jos esimerkiksi 100 ohmin vastuksen päiden välillä on 10 voltin jännite, voidaan vastuksen läpi kulkeva virta laskea taulukon 1.2 keskimmäisellä kaavalla seuraavasti:

```
I = U / R = 10 V / 100 ohmia = 0,1 A
```

Vastaavasti voidaan laskea kuinka suuri jännite pitää samaisen 100 ohmin vastuksen päiden välillä olla, jotta läpi kulkeva virta olisi 0,5 A.

Tähän käytetään taulukon 1.2 ensimmäistä kaavaa:

```
U = I x R = 0,5 A x 100 ohmia = 50 V
```

Jos tuntemattoman vastuksen päiden välillä on 25 V jännite ja sen läpi kulkee 0,2 A virta, voidaan vastuksen arvo laskea taulukon 1.2 kolmannella kaavalla:

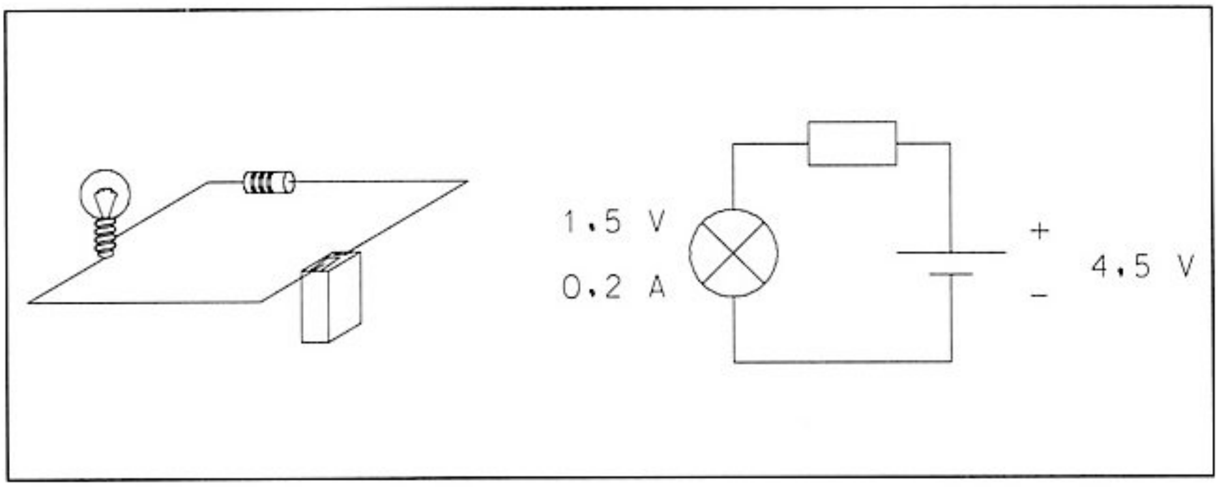
```
R = U / I = 25 V / 0,2 A = 125 ohmia
```

Esimerkki

Nyt kerätyillä tiedoilla voimme ratkaista yksinkertaisen käytännön ongelman:

Käytettävissä on lamppu, jossa lukee "1,5 V/0,2 A". Laatikon kätköistä löytyy vain litteä taskulampun paristo. Pariston kyljessä sanotaan jännitteeksi 4,5 volttia, joten lamppua ei voi kytkeä suoraan pariston napoihin, vaan väliin pitää kytkeä sopiva vastus. Millainen on sopiva vastus ?

Kytkentäkaavio on kuvan 1.2 mukainen. Lampun merkinnät tarkoittavat, että normaalikirkkaudella palaessaan lampun läpi kulkee 0,2 A virta ja että tällöin sen napojen välillä on 1,5 V jännite. Koska pariston jännite on 4,5 V, pitää väliin lisättävän vastuksen pudottaa jännitettä (4,5–1,5) V eli 3 V, jottei lamppu särkyisi.



Kuva 1.2. Yksinkertainen vastuskytkentä.

Ohmin lain mukaisesti vastuksen resistanssi on päiden välinen jännite jaettuna sen läpi kulkevalla virralla eli:

```
R = U / I = 3 V / 0,2 A = 15 ohmia
```

Kirchhoffin lait

Erityisesti aloittelevan rakentelijan on syytä tarkkaan harkita, mitä jännitteitä, virtoja ja vastuksia käyttää kaavoilla laskiessaan. Esimerkiksi edellä olleessa esimerkissä tulee jännitteenä käyttää vastuksen päiden välistä jännitettä eikä esimerkiksi pariston nimellisjännitettä 4,5 V tai lampun nimellisjännitettä 1,5 V, kuten voisi erehtyä tekemään. Tämänkaltaisia ongelmia ratkottaessa auttavat aina kaksi yksinkertaista ns. Kirchhoffin sääntöä. Ensimmäinen niistä kuuluu:

Pisteeseen tulevien virtojen summa on yhtä suuri

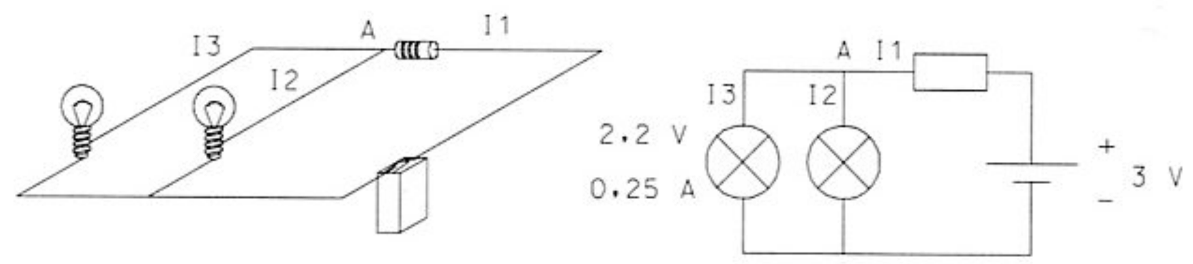
kuin siitä lähtevien virtojen summa.

Tätä kenties melko teoreettiselta kuulostavaa lausetta voi selventää esimerkillä.

Esimerkki

Kuvan 1.3 kytkentä on hieman vaikempi versio edellisestä tehtävästä. Tällä kertaa tulee saada kaksi samanlaista rinnakkain kytkettyä lamppua (2,2 V/0,25A) toimimaan 3 V paristolla. Jälleen tarvitaan etuvastus, jonka suuruus pitäisi ratkaista.

Pisteeseen A tulee paristosta vastuksen kautta virta, jota merkitsemme I1:llä. Samaan aikaan pisteestä A lähtevät lamppujen läpi kulkevat virrat I2 ja I3. Koska tiedämme lampuissa olevien merkintöjen ansiosta lamppujen läpi kulkevien virtojen oikeat arvot (I2 = I3 = 0,25 A), voimme laskea Kirchhoffin ensim-



Kuva 1.3. Kahden rinnakkain kytketyn lampun kytkentäkaavio.

mäisen säännön mukaisesti: pisteeseen A tulevien virtojen summa = siitä lähtevien virtojen summa:

```
I1 = I2 + I3 = 0,25 A + 0,25 A = 0,5 A
```

Pariston ja etuvastuksen läpi kulkee siis 0,5 A virta.

Kirchhoffin toinen laki kuuluu:

Virtalähteen lähdejännite on yhtä suuri kuin piirissä esiintyvien jännitehäviöiden summa.

Tarkastellaan jälleen kuvaa 1.3. Virtalähteen (pariston) lähdejännite on 3 V, ja lamppujen päiden välillä jännite on 2,2 V. Koska lähdejännite = jännitehäviöiden summa, voidaan vastuksen päiden välinen jännite laskea:

```
Uvastus = Uparisto - Ulamput = 3 V - 2,2 V = 0,8 V
```

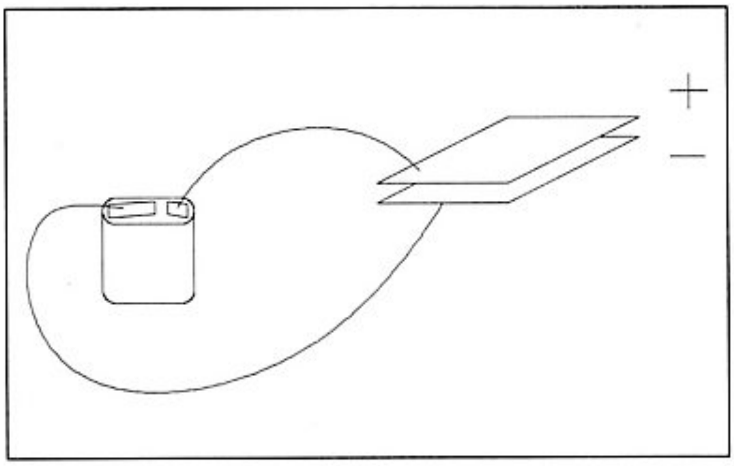
Nyt tiedämme siis sekä vastuksen läpi kulkevan virran suuruuden että sen päiden välisen jännitteen, joten vastuksen arvo ratkeaa helposti ohmin lain avulla:

```
R = U / I = 0,8 V / 0,5 A =  1,6 ohmia
```

Kapasitanssi

Kapasitanssiksi kutsutaan sähköisen osan kykyä varastoida sähkövaraus itseensä. Jokainen meistä on varmaan kokenut omaa sähkönvarauskykynsä käveltyään esimerkiksi kumipohjaisilla kengillä kokolattiamatolla ja saatuaan sitten mojovan sähköiskun ovenkahvaa tai jotain muuta metalliesinettä koskettaessaan.

Kapasitanssia saadaan aikaan, kun kaksi sähköä johtavaa kappaletta tuodaan lähelle toisiaan niin, ettei niiden välille kuitenkaan synny sähköistä yhteyttä.



Kuva 1.4. Metallilevyjen välille muodostuu kapasitanssi.

Kuvassa 1.4 esimerkiksi kaksi metallilevyä on asetettu lähelle toisiaan niin, että niiden väliin jää ohut ilmakerros.

Kun paristo kytketään kuvan 1.4 mukaisesti levyihin, virtaa pariston miinusnavasta elektroneja alempaan levyyn. Tällöin alempaan levyyn syntyy elektronien "ylijäämä". Vastaavasti pariston plusnapa imee elektroneja ylemmästä levystä, jolloin siihen syntyy elektronien "alijäämä". Sanotaan, että levyt ovat varautuneet, tai että niillä on sähkövaraus.

Jos paristo nyt irroitetaan, jää varaus edelleen levyihin. Kun varatut levyt yhdistetään johtimella, varaus purkautuu alalevyn ylijäämäelektronien virratessa ylälevyn vajausta täyttämään.

Mitä suurempia levyt ovat ja mitä lähempänä ne ovat toisiaan, sitä enemmän niihin voidaan varata sähköä ja sitä suurempi on niiden kapasitanssi. Kapasitanssin symboli on C. Yksikön nimi on faradi (F) ja kapasitanssia aikaansaavaa komponenttia kutsutaan kondensaattoriksi. Koska kondensaattorin levyt eivät ole toisissaan kiinni, ei virta kulje sen läpi.

Induktanssi

Monet ovat varmaan joskus rakentaneet sähkömagneetin kiertämällä eristettyä sähköjohdinta esimerkiksi rautanaulan ympärille. Sähkövirralla on sellainen mielenkiintoinen ominaisuus, että johtimen ympärille muodostuu aina magneettikenttä, kun siinä kulkee virtaa. Sama toimii myös toisinpäin, eli kun johdin liikkuu magneettikentässä tai kun johtimen ympärillä oleva magneettikenttä muuttuu, syntyy johtimeen jännite.

Kun johdin kierretään kelalle, osuvat jokaisen johdinsilmukan muodostamat yksittäiset magneettikentät rinnakkain vahvistaen toisiaan. Näin esimerkiksi kiertämällä johtimeen sata rinnakkaista kierrosta saadaan syntyneellä johdinkelalla aikaan satakertainen magneettikenttä suoraan johtimeen verrattuna. Kelan sisälle asetettu rautasydän edelleen vahvistaa kenttää.

Kelan ympärille muodostuvan magneettikentän luominen vaatii aina energiaa. Kun kelan läpi aletaan syöttää virtaa, varastoituu osa sähköenergiasta tuohon kenttään. Jos virtaa suurennetaan, magneettikentän voimakkuus kasvaa. Tämä ilmenee siten, että kun virtaa yritetään kasvattaa nopeasti, kela pyrkiikin hidastamaan muutosta. Hidastuminen johtuu juuri siitä, että osa energiasta jälleen varastoituu voimistuneeseen kenttään.

Vastaavasti jos virtaa pienennetään, heikkenee myös kelan magneettikenttä. Kentän muutos kuitenkin aiheuttaa kelassa jännitteen, joka pyrkii kasvattamaan kelan läpi menevää virtaa. Tämän vuoksi kestää hetken ennenkuin virta pienenee lopulliseen arvoonsa.

Kela pyrkii siis aina vastustamaan virrassa tapahtuvia muutoksia. Tasaisen voimakas virta kulkee kelasta läpi helposti, mutta virran muutokset aiheuttavat kelassa aina muutosta hidastavan ilmiön. Tätä kelan ominaisuutta sanotaan induktanssiksi. Induktanssin symboli on L ja yksikkö henry (H).

Vaihtosähkö

Tähän asti olemme rajoittuneet tilanteeseen, jossa sähkövirran suunta pysyy jatkuvasti samana. Hyvin tavallinen tilanne käytännössä on kuitenkin se, että virran suunta muuttuu jatkuvasti. Tällaista virtaa sanotaan vaihtovirraksi erotuksena jatkuvasti samaan suuntaan kulkevasta tasavirrasta. Vaihtovirran lisäksi puhutaan vaihtojännitteestä tai yleisesti vain vaihtosähköstä. Vaihtosähköä käytetään esimerkiksi joka kotiin tulevassa sähköverkossa.

Vaihtosähkö on siitä inhottava ilmiö, että monien asioiden laskeminen tulee paljon vaikeammaksi kuin tasasähköllä. Esimerkiksi Ohmin laki ei aina vaihtosähkölaitteissa olekaan voimassa.

Suureiden etuliitteet

Usein käytännön tilanteissa törmätään sellaisiin suureiden arvoihin, jotka ovat joko hirvittävän suuria tai pieniä. Esimerkiksi käytännön kytkennöissä käytetään usein suuruudeltaan 0,000 000 000 001 faradin kondensaattoreita ja 1 000 000 ohmin vastuksia.

Tällaiset pitkät nollajonot ovat hankalia käsitellä ja muistaa. Tämän vuoksi ne korvataan suureiden eteen tulevilla etuliitteillä. Esimerkiksi luku miljoona (1 000 000) voidaan korvata etuliitteellä "mega". Niinpä 1 000 000 ohmin vastusta kutsutaan yleensä yhden megaohmin vastukseksi. Vastaavasti luku

Etuliite	Lyhenne	Arvokerroin
mega	M	1 000 000
kilo	k	1 000
milli	m	0,001
mikro	μ	0,000 001
nano	n	0,000 000 001
piko	p	0,000 000 000 001

Taulukko 1.3. Elektroniikassa yleisesti käytettävät etuliitteet

0,000 000 000 001 voidaan korvata etuliitteellä "piko", joten em. kondensaattoria voidaan kutsua yhden pikofaradin kondensaattoriksi.

Etuliitteillä on kullakin myös oma lyhenteensä, jota käytetään silloin kun itse yksikkökin on lyhennetty. Kuten aiemmin todettiin, yksikkö faradi lyhennetään kirjaimella F. Etuliite piko taas lyhennetään kirjaimella p, joten pikofaradin lyhenne on yksinkertaisesti "pf".

Asia on varmasti jokaiselle tuttu muista jokapäiväisemmistä suureista. Esimerkiksi litra vettä on massaltaan 1 000 grammaa tai yleisemmin yhden kilogramman. Tulitikun halkaisija on noin 0,002 metriä tai tutummin 2 millimetriä. Etuliitteet kilo ja milli korvaavat lukuja 1 000 ja 0,001.

Etuliitteet piko ja mega edustavat elektroniikassa yleisesti tarvittavien etuliitteiden ääripäitä. Niiden väliin mahtuu jo mainittujen millin ja kilon lisäksi myös muita. Nämä on esitelty taulukossa 1.3. Taulukosta puuttuvat kymmentä ja sataa vastaavat etuliitteet (deka ja hehto), koska niitä ei elektroniikassa yleensä käytetä.

Muita etuliitteitä pyritään käyttämään siten että suureen lukuarvo pysyisi nollan ja tuhannen välissä. Niinpä on suositeltavampaa sanoa "100 kilo-ohmia" kuin "0,1 megaohmia".

2. Käytännön passiiviset komponentit

Tässä kappaleessa tutustumme tarkemmin niihin komponentteihin, joilla käytännössä saadaan aikaan edellisessä kappaleessa esitettyjä ilmiöitä, kuten resistanssia, kapasitanssia ja induktanssia.

Vastukset

Vastuksia käytetään elektronisissa kytkennöissä rajoittamaan virtojen ja jännitteiden suuruuksia. Niitä valmistetaan useista materiaaleista. Pienikokoiset vastukset ovat useimmiten erityisestä hiilimassasta muovattuja tankoja. Tangon koostumusta, kokoa ja muotoa muuttamalla tehtaassa valmistetaan halutun arvoisia vastuksia. Näitä vastuksia sanotaan massavastuksiksi ja ne ovat kaikista vastuskomponenteista halvimpia.

Hieman kalliimpia ovat ns. hiilikalvovastukset, jotka valmistetaan päällystämällä keraaminen tanko ohuella hiilikalvolla. Keraaminen runko voidaan päällystää myös ohuella metallikalvolla, jolloin kyseessä on ns. metallikalvovastus. Jos keraamisen rungon päälle metallikalvon tilalle kierretään vastuslangasta kela, saadaan lankavastus.

Vastukset voivat olla myös säädettäviä. Säätövastukset eli potentiometrit valmistetaan asettamalla vastuslangan tai kalvon pinnalle liikkuva metallikosketin, jota voi siirtää eri kohtiin säätöakselia kiertämällä. Säätövastuksia käytetään esimerkiksi äänentoistolaitteissa äänenvoimakkuuden ja -värin säädössä.

Vastusarvon toleranssi

Käytännön komponentteja valmistettaessa ei koskaan päästä täsmälleen haluttuun vastusarvoon. Tämän vuoksi kullekin vastustyypille määritellään rajat, joiden välissä arvo saa vaihdella. Näitä rajoja kutsutaan vastusarvon toleranssiksi ja ne ilmoitetaan yleensä prosentteina varsinaisesta nimellisarvosta.

Jos tehdas lupaa esimerkiksi 1 000 ohmin vastukselle 5 % toleranssin, voi vastuksen arvo vaihdella 5 % molemmin puolin 1 000 ohmia. Rajat ovat siis:

```
alaraja = 1000 ohmia - 1000 ohmia x 5/100 =950 ohmia
yläraja = 1000 ohmia + 1000 ohmia x 5/100 =1 050 ohmia
```

Tavanomaisiin kytkentöihin riittää yleensä jopa 20 % tarkkuus vastuksen arvon suhteen, mutta nykyään monet komponenttimyymälät myyvät ainoastaan 5 % ja vieläkin tarkempia vastuksia. Toisaalta vastukset ovat hinnaltaan kaikista elektroniikan komponenteista melkein halvimpia, joten liian tarkan vastuksen käytöstä ei juurikaan synny ylimääräisiä kustannuksia.

Vastuksen tehonkesto

Vastuskomponenteilla on aina ns. nimellisteho, jonka ylittäminen aiheuttaa komponentin tuhoutumisen. Vastuksen kuluttama teho lasketaan edellisen kappaleen taulukon 1.1 kaavoilla.

Esimerkki

Jos tuntemattoman vastuksen päiden välillä on 12 V jännite ja sen läpi kulkee 0,05 A virta, voidaan vastuksen kuluttama teho laskea taulukon 1.1 ensimmäisellä kaavalla:

```
P = U * I = 12 V x 0,05 A = 0,6 W
```

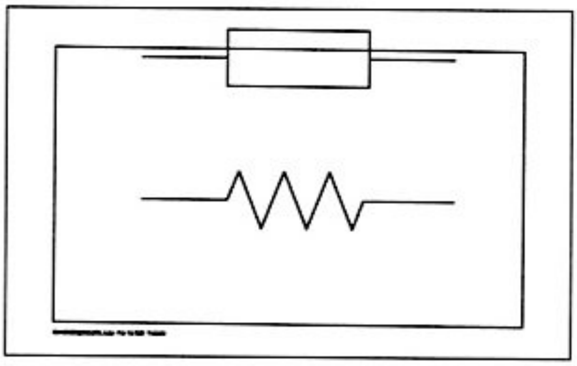
Vastuskomponentin nimellistehon pitäisi siis olla vähintään 0,6 W, jotta sitä ei ylikuormitettaisi. Jos 100 ohmin vastuksen läpi kulkee 0,03 A virta, voidaan vastuksen pienin riittävä nimellisteho laskea taulukon 1.1 keskimmäisellä kaavalla:

```
P = R x I x I = 100 ohmia x 0,03 A x 0,03 A = 0,09 W
```

Massa- ja kalvovastuksien nimellistehot vaihtelevat yleensä 0,25 watista pariin wattiin. Tehonkestoltaan suuremmat vastukset on yleensä toteutettu lankavastuksina.

Vastusten värikoodaus

Massa- ja kalvovastukset ovat yleensä kuvan 2.1 näköisiä tankoja, joiden päihin on yhdistetty metalliset liitosjohtimet. Vastusten ominaisuudet merkitään komponenttiin tavallisesti kolmella, neljällä tai viidellä värirenkaalla, joiden



Kuva 2.1. Massa- ja kalvovastusten kaavakuva.

merkitys on ennalta sovittu. Renkaiden tulkitseminen aloitetaan aina siitä päästä, jossa ne ovat lähempänä reunaa.

Kolmen renkaan vastuksessa renkaista kaksi ensimmäistä ilmoittavat vastusarvon kaksi ensimmäistä numeroa ja ne tulkitaan taulukon 2.1 mukaan. Kolmas rengas ilmoittaa noiden kahden luvun perään lisättävien nollien määrän saman taulukon mukaisesti. Kolmirenkaisen vastuksen toleranssi on 20%.

Neljän renkaan vastus tulkitaan muuten samoin kuin kolmirenkainen, mutta renkaista neljäs ilmaisee toleranssin. Tähän käytetään taulukkoa 2.2.

Musta	**0**
Ruskea	**1**
Punainen	**2**
Oranssi	**3**
Keltainen	**4**
Vihreä	**5**
Sininen	**6**
Violetti	**7**
Harmaa	**8**
Valkoinen	**9**

Jos renkaita on neljä tai viisi, ilmaisevat niistä vastaavasti kaksi tai kolme ensimmäistä vastusarvon kaksi tai kolme ensimmäistä numeroa ja toiseksi viimeinen rengas perään lisättävien nollien määrän.

Jos renkaita on kolme ilmaisevat niistä kaksi ensimmäistä vastusarvon kaksi ensimmäistä numeroa ja viimeinen niiden perään lisättävien nollien määrän saman taulukon mukaisesti.

Jos lukuarvon perään lisättävien nollien määrää ilmaiseva rengas on väriltään hopea tai kulta, ei ensimmäisistä renkaista saatuun lukuun lisätä nollia, vaan se jaetaan kymmenellä (hopea) tai sadalla (kulta).

Taulukko 2.1. Vastusten värikoodaus, ensimmäiset renkaat.

Vihreä	**0,5 %**
Ruskea	**1 %**
Punainen	**2 %**
Kulta	**5 %**
Hopea	**10 %**

Jos renkaita on kaikkiaan vähintään neljä, ilmaisee viimeinen niistä vastuksen toleranssin oheisen taulukon mukaisesti.

Jos renkaita on kaikkiaan kolme, on vastuksen toleranssi 20 %

Taulukko 2.2. Vastusten värikoodaus, viimeinen rengas.

Viiden renkaan vastuksessa kolme ensimmäistä rengasta ilmaisevat vastusarvon kolme ensimmäistä numeroa (taulukko 2.1), neljäs perään lisättävien nollien määrän (taulukko 2.1) ja viides toleranssin (taulukko 2.2).

Kaikilla rengasmäärillä loppuun lisättävien nollien määrän ilmaiseva rengas voi olla väriltään myös hopeinen tai kulta. Tällöin luvun perään ei lisätäkään nollia, vaan lukuarvo jaetaan kymmenellä (hopea) tai sadalla (kulta).

Esimerkkejä

1) Vastuksen ympärillä on kolme rengasta:

ruskea–musta–kulta

Kaksi ensimmäistä rengasta ilmaisevat taulukosta 2.1 luettavat vastusarvon kaksi ensimmäistä numeroa, ruskea = 1 ja musta = 0, siis "10". Kolmas rengas ilmaisee perään lisättävien nollien määrän. Kulta on kuitenkin erikoinen väri, joka aiheutti lukuarvon jakamisen kymmenellä, eli vastusarvoksi saadaan lopulta 1 ohmi. Koska renkaita oli vain kolme, on vastuksen toleranssi 20 %.

2) Vastuksen ympärillä on neljä rengasta:

ruskea–musta–punainen–kulta

Ensimmäiset kaksi rengasta ilmaisevat vastusarvon kaksi ensimmäistä numeroa taulukon 2.1 mukaisesti, eli ruskea = 1, musta = 0, siis "10". Kolmas rengas ilmoittaa taulukon 2.1 mukaisesti, että perään lisätään kaksi nollaa, vastusarvoksi tulee siis 1000 ohmia. Neljäs rengas ilmaisee taulukon 2.2 mukaisesti vastuksen toleranssin olevan 5 %.

3) Vastuksen ympärillä on viisi rengasta:

ruskea–musta–musta–ruskea–ruskea

Ensimmäiset kolme rengasta muutetaan suoraan numeroiksi taulukon 2.1 mukaan, eli ruskea = 1, musta = 0, musta = 0, siis "100". Neljäs rengas ilmoittaa lisättävien nollien määrän, tässä tapauksessa siis ruskea = 1. Vastusarvo on siis 1 000 ohmia. Viides rengas ilmaisee vastusarvon toleranssin, jolloin ruskea vastasi 1%:a. Kyseessä on siis tarkkuudeltaan 1% 1 000 ohmin vastus.

Kondensaattorit

Kondensaattoreilla on elektronisissa kytkennöissä useita hyvin erilaisia tehtäviä. Suurikapasitanssisia kondensaattoreita käytetään esimerkiksi tasoittamaan tasavirtajohtimissa esiintyviä jänniteheilahduksia. Kondensaattoreita valmistetaan vielä useammilla tavoilla kuin vastuksia. Kaikki ovat kuitenkin rakenteel-

taan periaatteessa samanlaisia, kaksi johtavaa levyä on asetettu vierekkäin eristävällä kerroksella toisistaan erotettuna.

Levyjen keskinäinen sijainti on myös hyvin vaihtelevaa, ne voidaan asettaa esimerkiksi kuvan 1.4 mukaisesti suoraan toisiaan vasten. Toinen tapa on leikata levyt pitkiksi suikaleiksi ja kierittää sitten rullalle. Näin saadaan hyvinkin suuriarvoinen kondensaattori kohtuullisen pieneen tilaan.

Karkeasti ottaen kondensaattorit voidaan valmistusmateriaalinsa puolesta jakaa kahteen pääryhmään, kuiviin ja elektrolyyttikondensaattoreihin.

Kuivat kondensaattorit

Kuivat kondensaattorit jaetaan eristyskerroksen valmistusmateriaalin mukaan luokkiin, joista yleisimmät ovat keraamiset, paperi- ja muovikondensaattorit. Kaikissa näissä nimi ilmaisee suoraan eristeenä käytetyn materiaalin.

Keraamiset kondensaattorit ovat arvoltaan kohtalaisen pieniä, alkaen pikofaradeista ja loppuen muutamaan sataan nanofaradiin. Paperikondensaattorit ovat nykyään jo melko harvinaisia niiden suuren koon ja muiden haittapuolien vuoksi.

Muovikondensaattorit lienevät nykyisistä kuivista kondensaattoreista yleisimpiä. Tähän ryhmään kuuluvat komponentit valmistetaan esimerkiksi polyesteristä, polykarbonaatista, styroflexista ja polystyreenistä.

Elektrolyyttikondensaattorit

Elektrolyyttikondensaattoreiden rakenne on hieman erikoinen muihin verrattuna. Niissä nimittäin toisen levyn muodostaa alumiinipinta ja toisen sitä koskettava elektrolyyttineste. Neste on koostumukseltaan sellaista, että sähkövirta saa aikaan kemiallisen reaktion, joka muodostaa alumiinin pintaan ohuen eristävän oksidikerroksen.

Elektrolyyttikondensaattorit ovat muihin tyyppeihin verrattuina suurikapasitanssisia, sadoista nanofaradeista aina satoihin tuhansiin mikrofaradeihin.

Säätökondensaattorit

Kondensaattorin kapasitanssiin vaikuttaa levyjen koko sekä niiden välinen etäisyys. Tekemällä jompikumpi näistä säädettäväksi saadaan ns. säätökondensaattori, jonka kapasitanssia voidaan muuttaa. Säätökondensaattoreita käytetään yleisesti esimerkiksi radiovastaanottimissa vastaanottotaajuuden valintakomponenttina.

Kondensaattorin jännitteenkesto

Kaikilla kondensaattoreilla on valmistajan ilmoittama suurin sallittu jännite, jonka komponentin päiden välille saa asettaa. Jos se ylitetään, saattaa korkea jännite tunkeutua eristekerroksen läpi pienen salaman tavoin. Seurauksena tällaisesta läpilyönnistä on yleensä ainakin itse kondensaattorin tuhoutuminen, mikä usein puolestaan aiheuttaa laitteen muillekin komponenteille vahinkoa.

Suurin sallittu jännite on yleensä suurikokoisissa kondensaattoreissa merkitty komponentin kuoreen, pienemmillä kondensaattoreilla pitää turvautua valmistajan tuoteluetteloiden tietoihin.

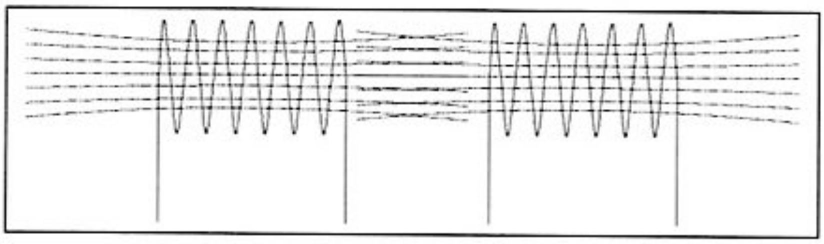
Jotkut kondensaattorit pitää kytkeä myös oikein päin jännitteen napaisuuteen nähden. Esimerkiksi useimpiin elektrolyyttikondensaattoreihin on selkeästi merkitty joko plus- tai miinusnavan sijainti.

Kelat ja muuntajat

Johdinkeloja käytetään pääasiassa kahteen tarkoitukseen. Koska kelan induktanssi vastustaa kelan läpi kulkevan virran muutoksia, voidaan kelaa käyttää tasaamaan ei-toivottuja jännitevaihteluita. Toinen tärkeä käyttökohde ovat erilaiset radiolaitteet, joissa kelat ovat tärkeässä asemassa ns. virityspiireissä.

Kun kaksi kelaa asetetaan kuvan 2.2 mukaisesti niin, että niiden magneettikentät joutuvat toistensa sisään, saadaan muuntaja. Kun muuntajan toiseen kelan läpi kulkee vaihtovirta, muuttuu sen magneettikenttä virran tahdissa. Jatkuvasti muuttuva magneettikenttä aiheuttaa toisen kelan päiden välille samaan tahtiin vaihtuvan vaihtojännitteen.

Toiseen kelaan muodostuvan jännitteen suuruutta voidaan muuttaa kelojen kierrosmääriä muuttamalla. Tästä ominaisuudesta koko komponentti onkin saanut nimensä.



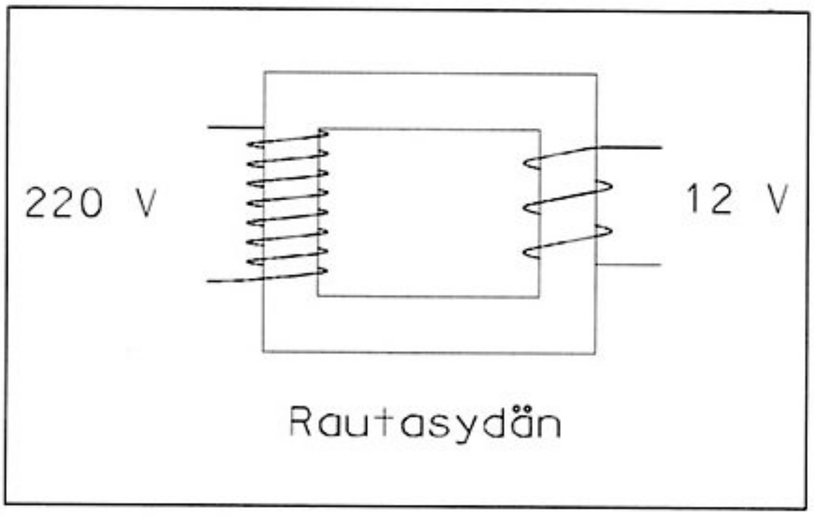
Kuva 2.2. Kelojen magneettikentät osuvat kohdakkain.

Kierrosmäärät on kaiken lisäksi helppo laskea, sillä kelojen kierrosmäärien suhde on yhtä suuri jännitteiden suhteiden kanssa, eli kaavana ilmaistuna:

```
M1 / M2 = U1 / U2
```

Tässä M1 ja M2 tarkoittavat ensimmäisen ja toisen kelan kierrosmääriä ja U1 ja U2 vastaaavia jänniteitä. Yleensä sitä muuntajan kelaa johon muunnettava jännite tuodaan, sanotaan ensiökäämiksi ja toista toisiokäämiksi (myös sekundäärikäämi). Käytännön muuntajissa magneettikentän siirtymistä vahvistetaan kuvan 2.3 mukaisella kelojen läpi kulkevalla rautasydämellä.

Muuntajia käytetään hyvin yleisesti mm. miltei kaikessa kodinelektroniikassa pudottamaan sähköverkon 220 voltin jännite elektroniikalle paremmin sopivaksi esimerkiksi 12 voltiksi (kuva 2.3).

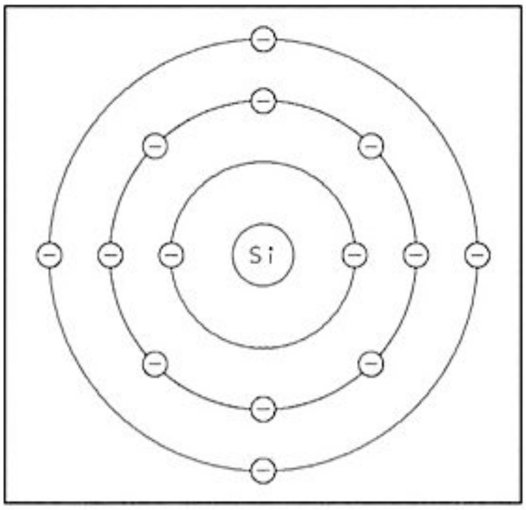


Kuva 2.3. Jännitteen pudottaminen muuntajalla.

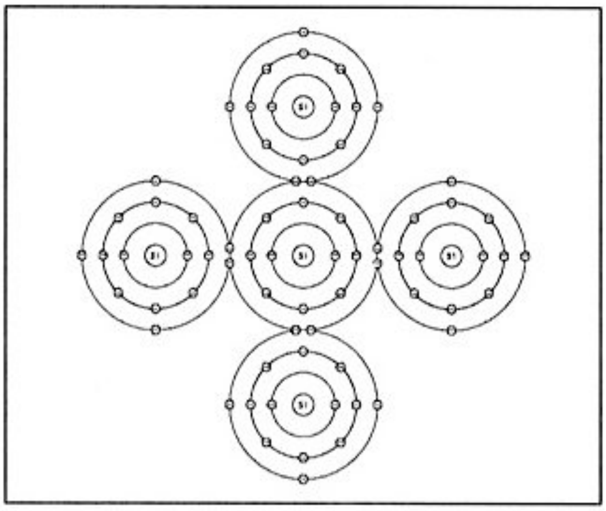
3. Puolijohteet

Puolijohteiksi sanotaan sellaisia aineita, jotka sijoittuvat sähkönjohtokyvyltään johteiden ja eristeiden välimaastoon. Tarkastellaan esimerkkinä vaikkapa ainetta nimeltään pii, kemialliselta lyhenteeltään Si. Pii on alkuaine, jonka atomin kaaviokuva on esitetty kuvassa 3.1. Uloimmalla kuorellaan sillä on neljä elektronia, joten uloin kuori on vain "puolillaan".

Kiinteässä piissä atomit ovat ryhmittyneet kuvan 3.2 (sisemmät elektronikuoret on jätetty pois) mukaisesti siten, että kukin atomi jakaa elektroninsa neljän muun atomin kanssa. Näin kunkin atomin uloin kuori on tavallaan täynnä.



Kuva 3.1. Piin atomin kaaviokuva.

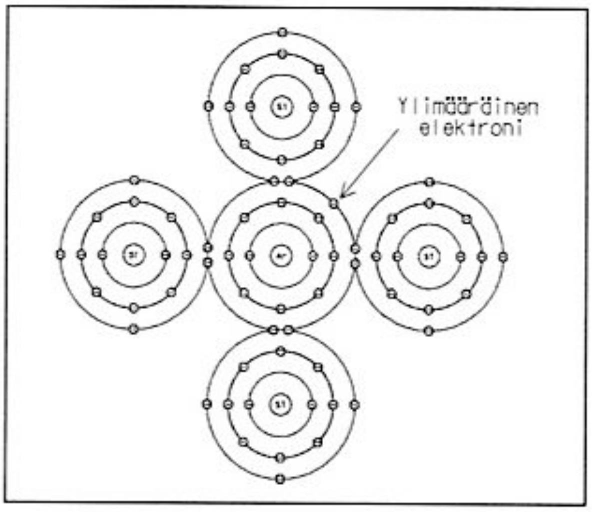


Kuva 3.2. Piin atomien ryhmittely.

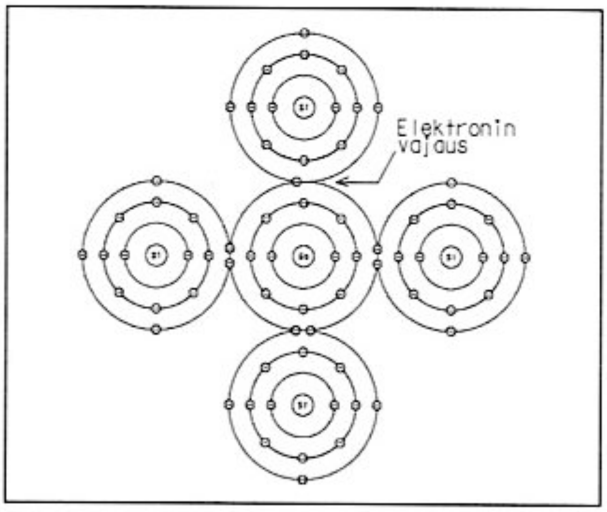
Puolijohteiden teoriaa

Jos puolijohteeseen sekoitetaan pieni määrä sopivaa toista alkuainetta, saadaan varsin mielenkiintoisia aineita. Esimerkiksi piille sopivia lisäaineita ovat arseeni (As) ja gallium (Ga). Näillä aineilla on uloimmalla kuorellaan piihin verrrattuna arseenilla yksi elektroni enemmän ja galliumilla yksi vähemmän.

Kun arseeniatomi asettuu piikiteessä piiatomin tilalle, muodostuu kuvan 3.3a mukainen tilanne. Arseenin uloimman kuoren neljä elektronia asettuvat piin tavoin, mutta viidennelle ei löydy paikkaa. Niinpä kiteeseen jääkin harhai-



Kuva 3.3a. N-tyypin puolijohde.



Kuva 3.3b. P-tyypin puolijohde.

lemaan ylimääräinen elektroni. Tällaista ainetta sanotaan N-tyypin puolijohteeksi, koska se johtaa sähköä negatiivisten varauksenkuljettajien (elektronien) ansiosta.

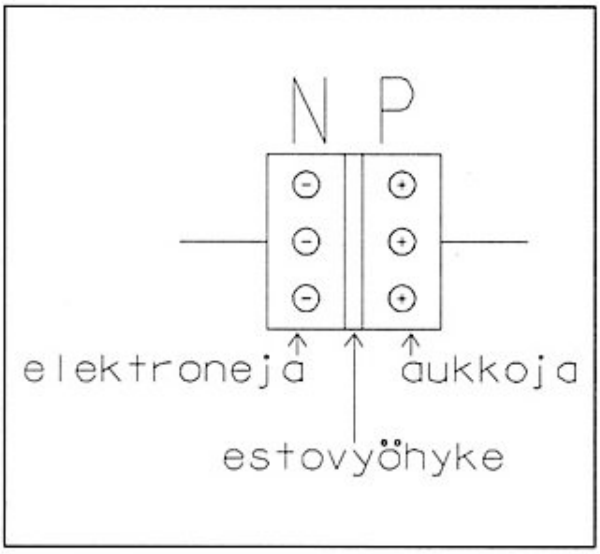
Vastaavasti voidaan piihin sekoittaa galliumia, jolloin kiderakenne on kuvan 3.3b mukainen. Nyt gallium pystyy jakamaan elektroneja vain kolmen ympäröivän atomin kanssa. Neljännen piiatomin ja galliumatomin välille jää yhden elektronin vajaus eli aukko. Tällaista ainetta sanotaan P-tyypin puolijohteeksi ja se johtaa sähköä, koska aineeseen ulkoa tulevat elektronit hyppivät helposti aukosta toiseen.

Puolijohteet ovat vielä sikäli erikoisia aineita, että niiden sähkönjohtokykyä voidaan rajusti muuttaa säätämällä jotain ulkoista ympäristötekijää kuten lämpötilaa tai puolijohteelle osuvan valon voimakkuutta.

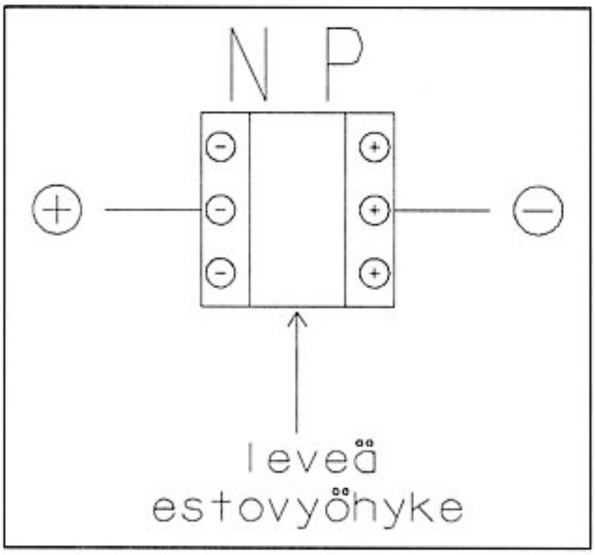
Diodi

Kun N- ja P-tyypin puolijohteista valmistetut palat asetetaan kuvan 3.4a mukaisesti vastakkain, saadaan komponentti nimeltään diodi. Kun puolijohdepalat yhdistetään, pyrkivät aluksi P-palan aukot siirtymään N-palaan ja N-palan vapaat elektronit P-palaan. Näin tapahtuukin hetken aikaa, mutta pian siirtyminen lakkaa, koska rajapinnan aukot ja elektronit korvaavat toisensa ensimmäisenä, eikä sen jälkeen rajapinnan luona enää ole sen enempää ylimääräisiä elektroneja kuin aukkojakaan varauksia kuljettamassa. Tätä muodostunutta eristävää aluetta kutsutaan estovyöhykkeeksi.

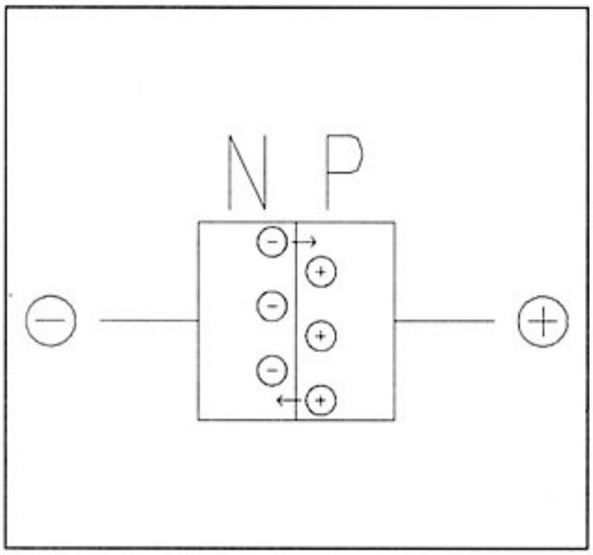
Jos diodi kytketään kuvan 3.4b mukaisesti jännitelähteeseen, pyrkii N-palaan tuotu positiivinen varaus imemään N-palan elektroneja pois diodista. Vas-



Kuva 3.4a. Diodi.



Kuva 3.4b. Diodi estosuunnassa.



Kuva 3.4c. Diodi päästösuunnassa.

taavasti P-palaan tuotu negatiivinen varaus pyrkii imemään P-palan aukkoja. Tämän vuoksi diodin estovyöhyke levenee entisestään, eikä virta pääse kulkemaan sen läpi. Diodi toimii siis tässä tapauksessa eristeenä.

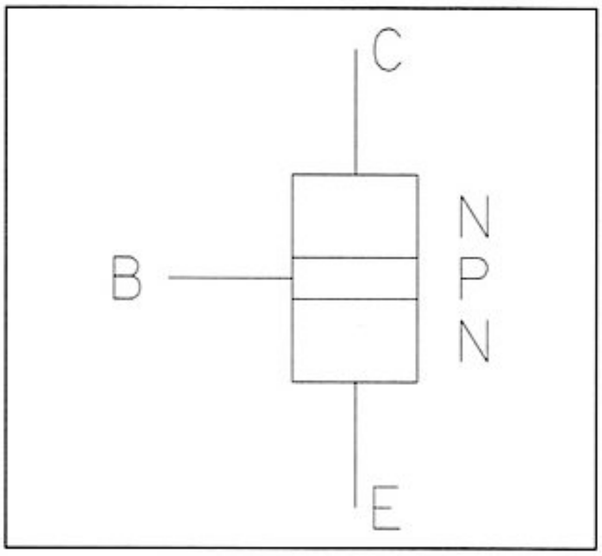
Jos jännite kytketään diodiin toisin päin (kuva 3.4c), diodiin tuodut varaukset puristavat estovyöhykettä kapeammaksi, kunnes riittävän suurella jännitteellä elektronit ja aukot pääsevät hyppimään rajapinnan yli. Tässä tapauksessa diodi toimiikin johteena.

Diodi on siis komponentti, joka päästää virran kulkemaan vain toiseen suuntaan. Jännitettä, jolla diodi myötäsuunnassa alkaa johtaa, sanotaan diodin kynnysjännitteeksi ja se on piidiodeilla noin 0,6–0,7 volttia.

Transistori

Rakennetaanpa seuraavaksi hieman monimutkaisempi komponentti. Tällä kertaa asetetaan kahden N-palan väliin yksi ohut P-pala kuvan 3.5 mukaisesti. Kuhunkin palaan liitetään johdin jotka on kuvassa nimetty kirjaimilla E, B ja C. Jos johtimien B ja E välille kytketään nyt jännite siten, että B kytketään positii-

viseen ja E negatiiviseen napaan, kulkee komponentin läpi virtaa. Tämä johtuu siitä, että alempi N-pala ja P-pala muodostavat myötäsuuntaan kytketyn diodin.



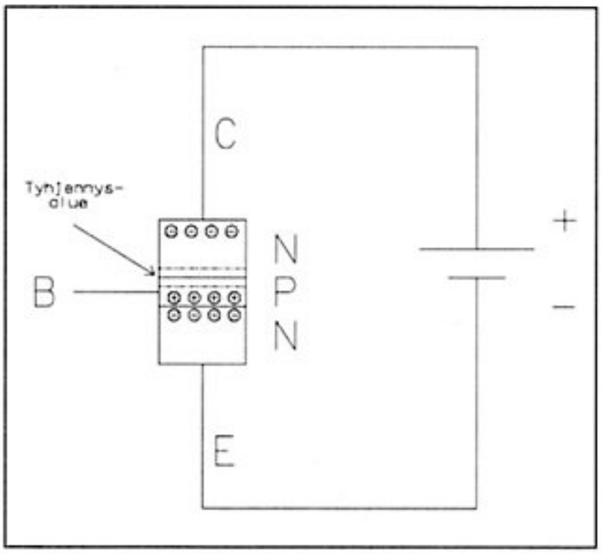
Kuva 3.5. Transistori.

Jos B-johtimen positiivinen jännite siirretään johtimeen C, ei virtaa enää kulje, koska ylemmän N-palan ja P-palan muodostama diodi on estosuunnassa.

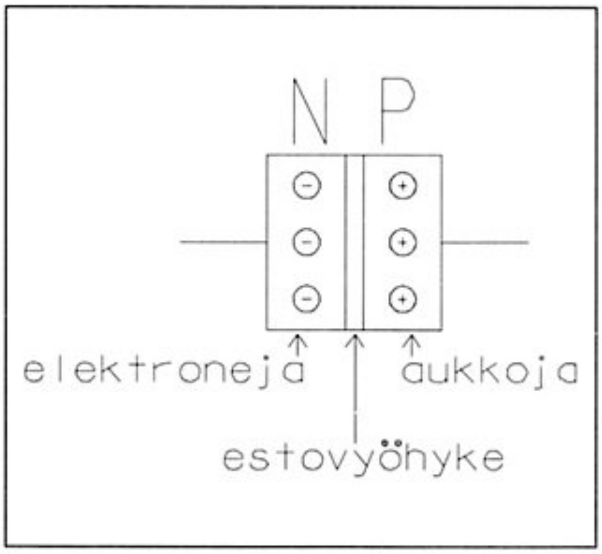
Entäpä jos positiivinen jännite kytketään sekä B- että C-johtimeen? Voisi kuvitella, että B:n kautta kyllä kulkee virtaa, mutta C:n kautta ei. Todellisuudessa kuitenkin tapahtuu varsin erikoinen ilmiö. Kun B-johtimelle tuodaan pienikin jännite, ryöpsähtää C-johtimestakin virta E:lle.

Miten tämä on mahdollista? Tarkastellaanpa tilannetta hieman tarkemmin. Kuvassa 3.6a on esitetty aukkojen ja elektronien asettuminen paloilla silloin, kun vain välille E–C on kytketty jännite. Koska väli C–B on estosuunnassa, muodostuu niiden väliin samanlainen estovyöhyke kuin estosuunnassa olevaan diodiin, eikä virta pääse kulkemaan.

Jos nyt myös välille E–B kytketään jännite kuvan 3.6b mukaisesti, alkavat elektronit kulkea myötäsuuntaan kytketyn E–B-rajapinnan yli. Jos keskellä oleva P-pala on kuitenkin riittävän ohut, käykin niin, että E-palasta suurella no-



Kuva 3.6a. Transistori sulkutilassa.



Kuva 3.6b. Transistori avaustilassa.

peudella kohti B:tä vilistävistä elektroneista vain pieni osa ehtii kääntyä kohti B-johdinta suurimman osan kulkiessa läpi suoraan C-palaan.

Kun elektronit ovat päässeet hankalan B-C-rajapinnan yli, ei mikään enää estäkään niiden kulkua, joten ne pääsevät etenemään C-palankin läpi johtimen kautta takaisin paristoon. Tällaista kolmesta päällekkäisestä puolijohdepalasta koottua komponenttia sanotaan transistoriksi.

Esimerkissä kokosimme palat alhaalta lukien järjestyksessä N, P ja N, mutta ne voidaan myös vaihtaa keskenään siten, että laidoille laitetaan P-palat ja keskelle ohut N-pala. Tässä tapauksessa kuitenkin myös käytettyjen jännitteiden suunnat pitää vaihtaa. Transistorit jaetaan palojen järjestyksen mukaan kahteen pääryhmään: NPN- ja PNP-transistoreihin.

Johtimille annetut nimet tulevat englanninkielisistä sanoista emitter (E), base (B) ja collector (C). Suomessa puhutaan vastaavasti emitteristä, kannasta ja kollektorista. Englanninkieliset nimet ovat melko loogisia, jos ajatellaan elektronien kulkua transistorissa. Kaikki transistorin läpi kulkevat elektronit tulevat ensin emitteriin joka "säteilee" ne muihin paloihin (engl. emit, säteillä). Kollektori taas "sieppaa" suurimman osan elektroneista (collect, kerätä).

Erittäin tärkeä piirre transistorin toiminnassa on se, että hyvin pienikin kantavirta saa aikaan suuren virran kollektorissa. Samoin pienikin muutos kantavirrassa aiheuttaa suuren muutoksen kollektorivirrassa. Tästä johtuu se, että transistoria voidaan käyttää virran vahvistimena. Tuodaan vain heikko virta transistorin kannalle, niin vahvistettu virta näkyy kollektorilla.

Transistorin vahvistuskykyä kuvataan usein suureella tasavirtavahvistuskerroin, lyhennettynä hFE. Vahvistuskertoimet vaihtelevat melko runsaasti käytännön komponenteilla alkaen noin kymmenestä kasvaen aina useaan sataan, jopa lähelle tuhatta.

4. Käytännön puolijohdekomponentit

Todellisia puolijohdekomponentteja hankittaessa ei pelkkä komponentin toiminnan ymmärtäminen riitä, sillä elektroniikan komponentteja on tarjolla huikeita määriä. Esimerkiksi erityyppisiä transistoreita löytyy melkein kaupasta kuin kaupasta satoja erilaisia.

Kaikilla komponenteilla on omat käyttötarkoituksensa ja rajoituksensa, jotka pitää osia hankkiessaan tuntea.

Diodi

Diodin tarkoitus on päästää virta kulkemaan yhteen suuntaan, mutta estää sen kulku toiseen suuntaan. Tämä on yhteistä useimmille diodeille. Eroja käytännön komponenteissa onkin tämän jälkeen runsaasti.

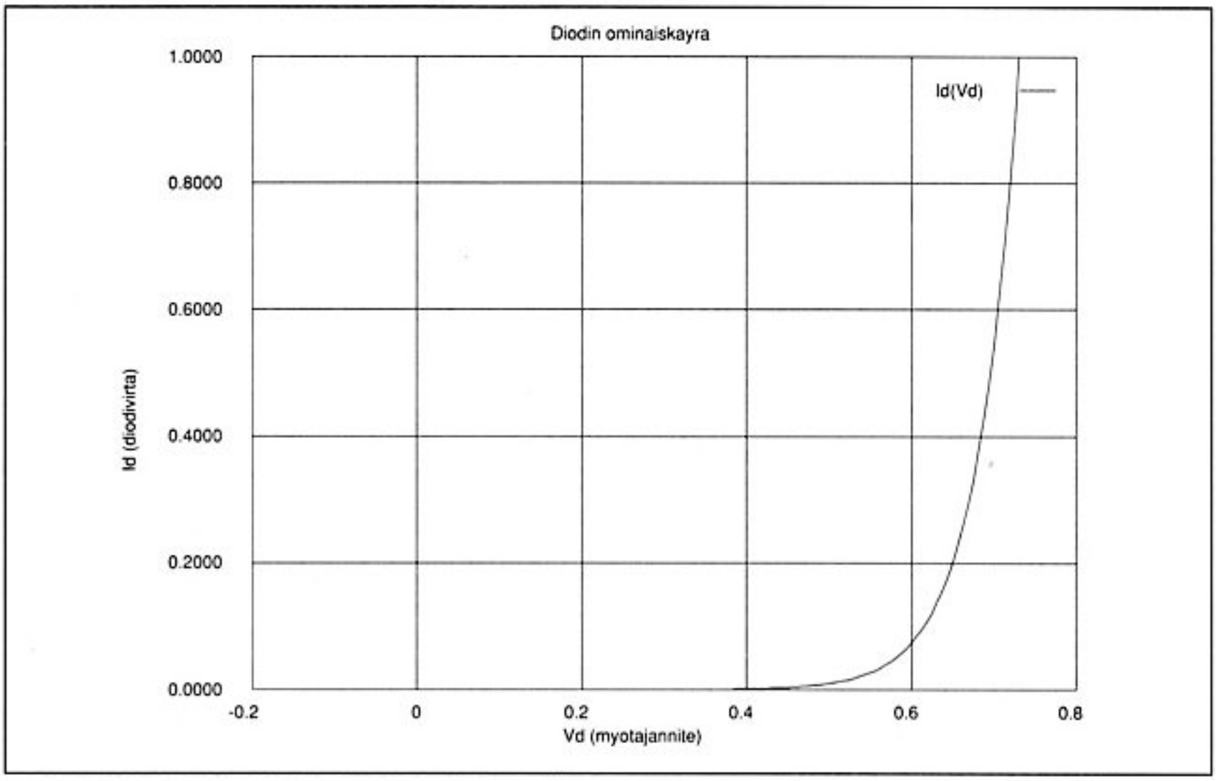
Ensimmäinen ratkaiseva tekijä on suurin sallittu diodin läpi kulkeva virta. Pienikokoinen germaniumdiodi sopii esimerkiksi radioantennista saatavan heikon, ehkä muutaman mikroampeerin virran tasasuuntaukseen, kun taas auton laturista saatavaa vaihtosähköä tasasuuntavan diodin täytyy kestää useiden kymmenien ampeerien virta.

Toinen tärkeä diodin käyttöä rajoittava tekijä on sen suurin sallittu estosuuntainen jännite. Jos diodin päiden välillä oleva jännite kasvaa liian suureksi, tapahtuu komponentissa läpilyönti, joka yleensä tuhoaa diodin.

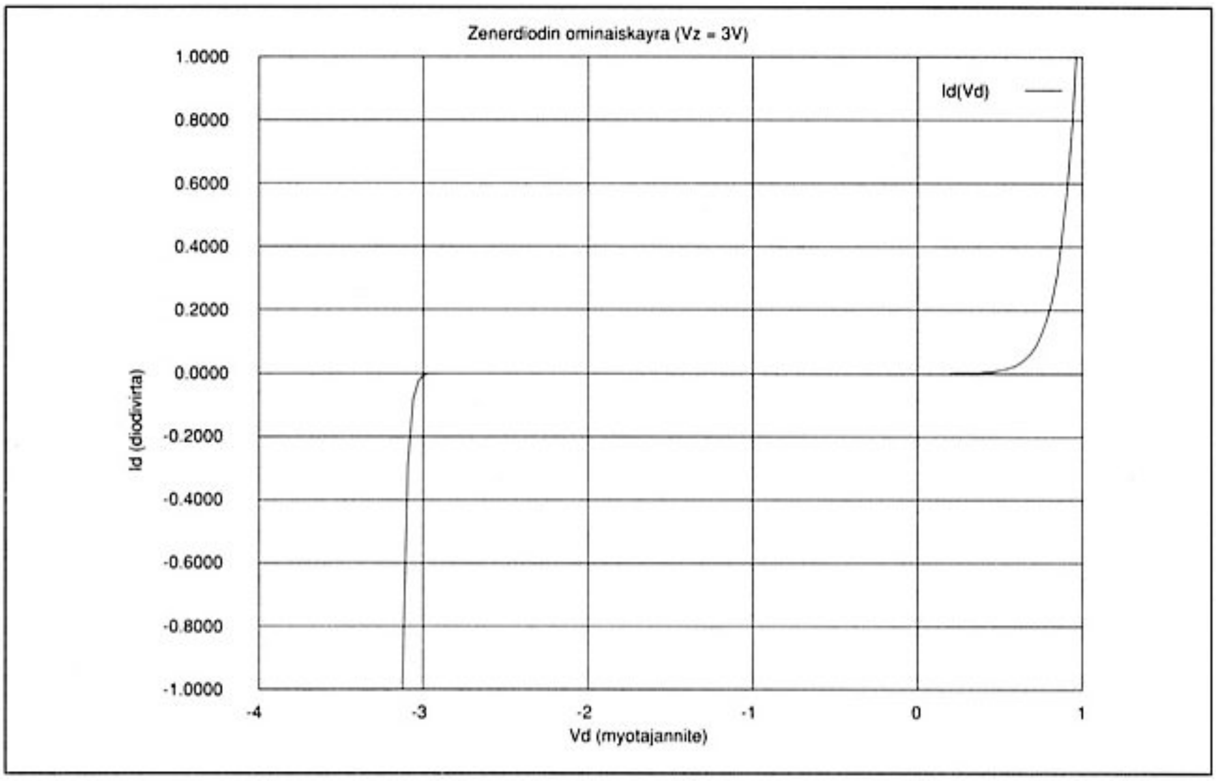
Virran suunnan kontrolloinnin lisäksi diodeja käytetään moniin muihinkin tarkoituksiin. Monia erikoistarkoituksia varten valmistetaan erikoisdiodeja, jotka poikkeavat sekä rakenteeltaan että toiminnaltaan muista.

Zenerdiodilla on sellainen erikoisominaisuus, että kun sen päiden välille kytketty estosuuntainen jännite nousee tietyn rajan yli, alkaa diodi yllättäen johtaa. Tätä jännitettä kutsutaan diodin zenerjännitteeksi. Zenerdiodeita käytetään pääasiassa jännitteiden tiettyyn arvoon vakavointiin.

Diodin ja zenerdiodin eroa selventävät kuvat 4.1 ja 4.2. Kuvissa esitetään käyrän avulla kuinka diodin läpi kulkevan virran suuruus muuttuu eri jännitteil-



Kuva 4.1. Piidiodin ominaiskäyrä.



Kuva 4.2. Zenerdiodin ominaiskäyrä.

Nimi	If (A)	Vr (V)	Kuvaus
1N4001	1	50	Yleiskäyttöinen tasasuuntausdiodi
1N4007	1	1 000	Suurjännitetasasuuntausdiodi
OA91	0,05	90	Piensignaali-ilmaisindiodi (germanium)
1N4148	0,08	75	Yleiskäyttöinen piensignaalidiodi
SSIE1140	35	700	Suurvirta- ja jännitetasasuuntausdiodi

Taulukko 4.1. Muutamia diodeja taulukoituna.

lä. Tavallisen diodin läpi kulkee virtaa vain kynnysjännitettä (kuvassa 4.1 noin 0,6 V) korkeammilla positiivisilla jännitteillä. Zenerdiodi puolestaan johtaa myös negatiivisen jännitteen noustessa yli zenerjännitteen (kuvassa 4.2 noin 3,0 V).

Zenerdiodeja on saatavana yleisesti zenerjännitteillä noin 3 voltista aina pariinsataan volttiin.

Diodin P- ja N-kerrosten välille jäävät vyöhykkeet muodostavat aina pieniä kondensaattoreita. Näiden kondensaattoreiden kapasitanssia on erityisesti vahvistettu kapasitanssidiodissa. Säätämällä kapasitanssidiodin päiden välillä olevaa jännitettä voidaan sen sisäistä kapasitanssia muuttaa. Tätä käytetään hyväksi esimerkiksi radio- ja tv-vastaanottimissa, joissa kapasitanssidiodeilla valitaan vastaanottotaajuus.

Muutamia diodeja on esitelty taulukossa 4.1.

Transistorit

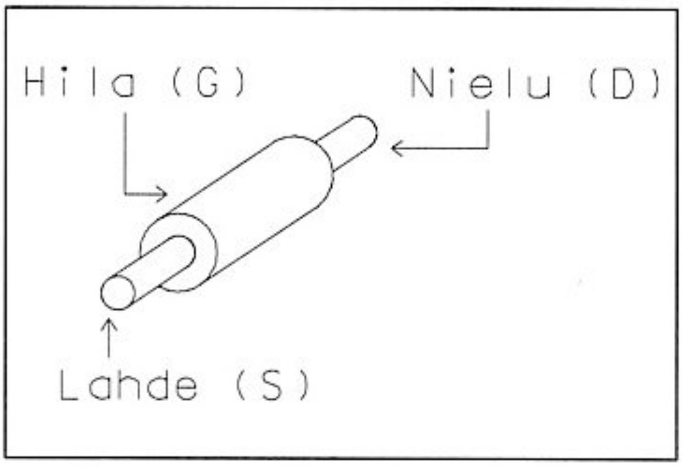
Transistoreilla on hyvin samankaltaisia ominaisuuksia kuin diodeilla. Diodin läpi kulkevan virran tavoin transistoreilla on omat maksimiarvonsa kaikkien kolmen navan kautta kulkeville virroille, samoin eri napojen välisille jännitteille.

Lisäksi transistoreiden tärkeisiin ominaisuuksiin kuuluu se, kuinka nopeasti komponentti pystyy vaihtamaan tilaansa. Tätä ominaisuutta kutsutaan transistorin taajuusvasteeksi.

Aivan kuten diodeja, myös transistoreita on ajan mittaan kehitelty useita tyyppejä. Perinteistä aiemmin kuvailtua transistoria kutsutaan bipolaaritransistoriksi.

Kanavatransistori eli FET (Field Effect Transistor) on rakenteeltaan bipolaaritransistorista poikkeava. Kolmen kerroksen sijasta kanavatransistorissa on yksi transistorin läpi kulkeva ohut kanava. Kanavan molempiin päihin on kytketty

emitteriä ja kollektoria vastaavat johtimet, joita kanavatransistorissa usein kutsutaan lähteeksi (source) ja nieluksi (drain).



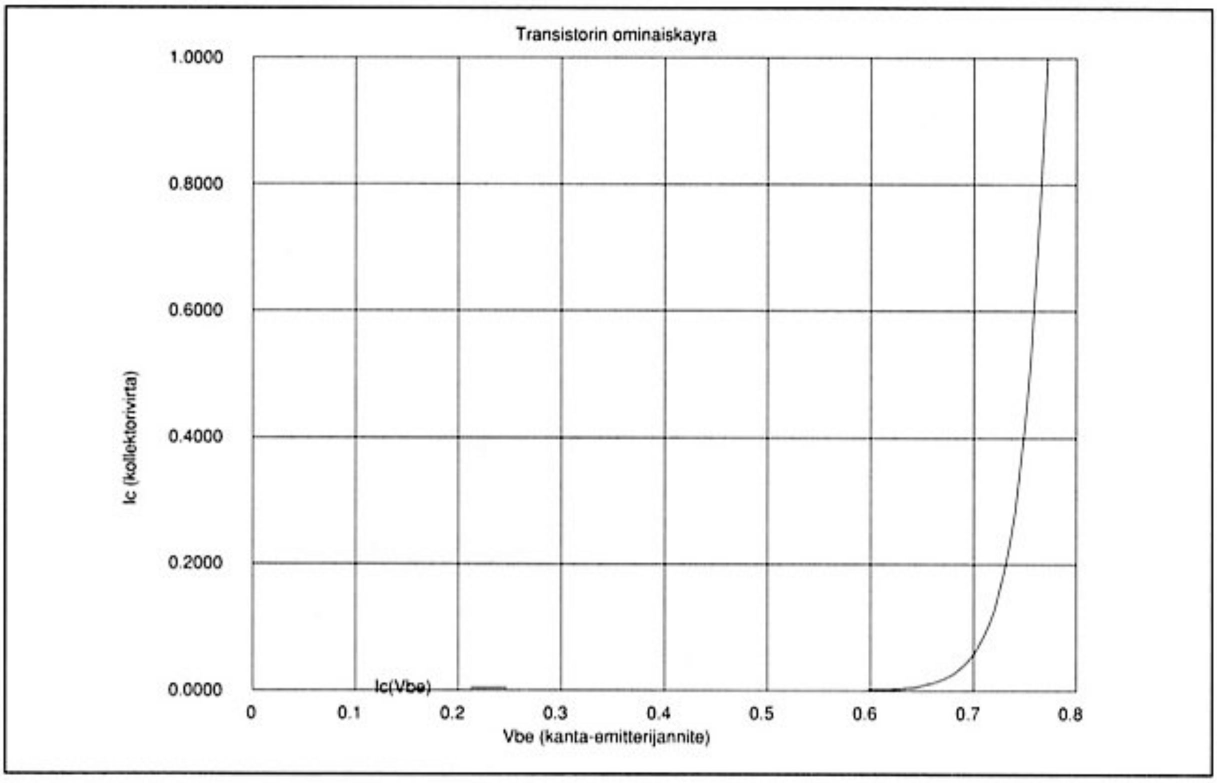
Kuva 4.3. Kanavatransistori.

Kanavaa ympäröi transistorin kolmanteen napaan hilaan (gate) kytketty kerros. Hilan jännitettä muuttamalla voidaan säätää kanavan läpi kulkevaa virtaa (kuva 4.3). Kanavatransistorin hilan läpi kulkee vain hyvin pieniä virtoja, mikä tekee siitä erityisen hyvän komponentin erittäin suurta herkkyyttä vaativiin laitteisiin.

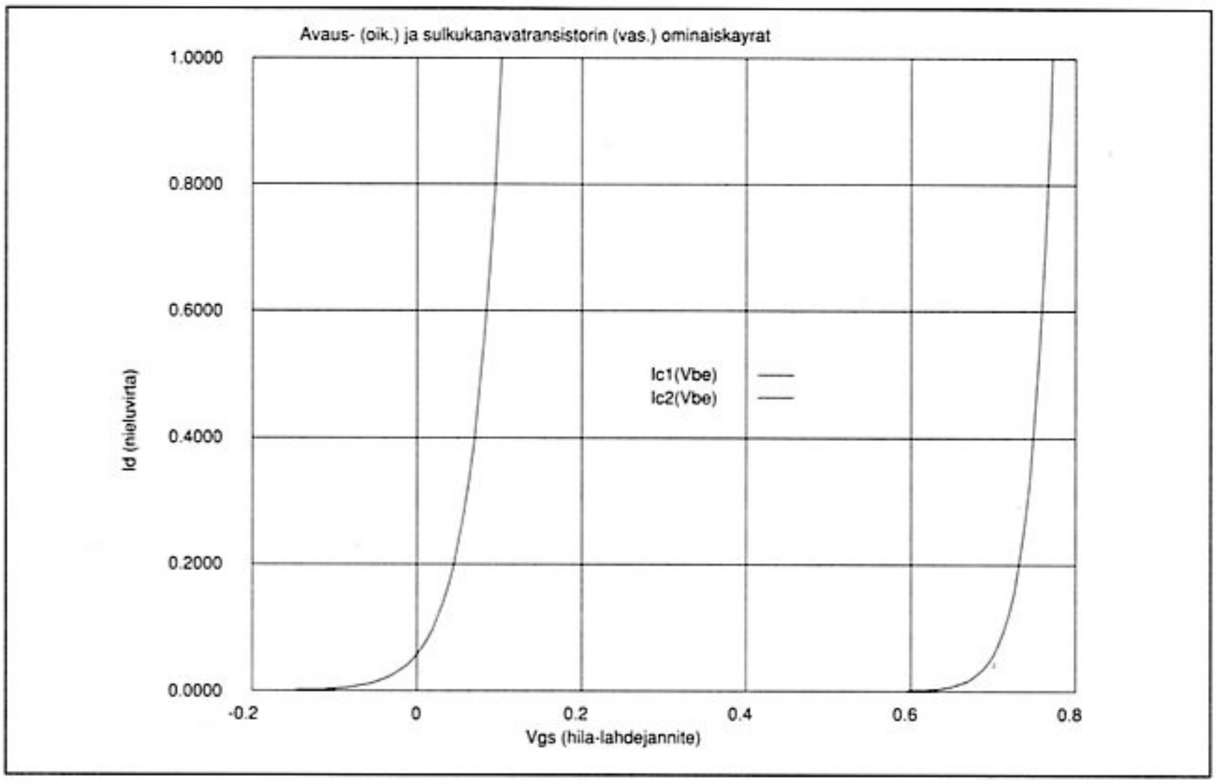
Kanavatransistorista on edelleen kehitetty useita versioita, esimerkiksi MOS-FET ja JFET, joilla kaikilla on omat hyvät ja huonot puolensa.

Kuvassa 4.4 on esitetty miten bipolaaritransistorin kollektorivirta riippuu kanta-emitterijännitteestä, kun emitteri-kollektorijännite pidetään vakiona.

Niinsanotun avaustyyppisen kanavatransistorin (enhancement FET) ominaiskäyrä on hyvin samantyyppinen, mutta toinen, sulkutyyppinen (depletion FET)



Kuva 4.4. Bipolaaritransistorin kollektorivirran riippuvuus kanta-emitterijännitteestä.



Kuva 4.5. Kanavatransistoreiden ominaiskäyrät.

Nimi	Ic (A)	Vceo (V)	Ptot (W)	fT (MHz)	nap.	hFE
2N2222	0,8	30	0,5	250	NPN	200
	Yleiskäyttöinen piensignaalitransistori					
BD235	2	60	25	20	NPN	25
	Yleiskäyttöinen tehotransistori					
TIP51	5	250	100	20	NPN	10
	Nopea tehotransistori					
BFY76	1	60	0,4	100	NPN	150
	Pienikohinainen piensignaalitransistori					
BD676	2	45	40	20	PNP	750
	Voimakkaasti vahvistava yleistransistori					
Ic	= Suurin sallittu kollektorivirta					
Vceo	= Suurin sallittu kollektori-emitterijännite					
Ptot	= Suurin sallittu tehohäviö transistorissa					
fT	= Suurin toimintataajuus					

Taulukko 4.2. Muutamia bipolaaritransistoreita taulukoituna.

käyttäytyy sikäli eri tavoin, että virta kulkee sen läpi jopa negatiivisilla hilajännitteillä (kuva 4.5).

Mikropiirit

Mikropiirillä tarkoitetaan sellaista kytkentää, joka on rakennettu valmistamalla yhdelle piipalalle useita komponentteja ja johdottamalla ne sopivaksi kytkennäksi. Mikropiirejä alettiin tehdä 60- ja 70-lukujen vaihteessa aluksi valmistamalla useita transistoreita samalle puolijohdepalalle ja myöhemmin lisäämällä mukaan myös muita komponentteja.

Yksittäisten mikropiirien toiminta riippuu täysin sen sisäisestä kytkennästä, joten niitä ei voi oikeastaan käsitellä kovin tarkasti yhtenä komponenttiryhmänä.

Nykyään mikropiirit jaetaan yleensä kahteen pääluokkaan: digitaalisiin ja analogisiin. Analogiset mikropiirit (ja kytkennät yleensä) käsittelevät ajan suhteen jatkuvia signaaleja, esimerkiksi tavanomaiset äänentoistolaitteet ovat hyvin tyypillistä analogiatekniikkaa.

Digitaaliset piirejä käsitellään tulevissa luvuissa tarkemmin.

5. Digitaalitekniikka

Nyt, kun olemme selvittäneet sähkötekniikan perusteita ja yksinkertaisten puolijohdekomponenttien toimintaa, olemme valmiita siirtymään kohti varsinaista digitaalielektroniikkaa, eli juuri sitä elektroniikan alaa, joka tietokoneiden syövereissä jyllää.

Digitaalikytkentöjen mukava puoli on siinä, ettei niitä tehdessä tarvitse juurikaan ajatella, miten suuria virtoja tai jännitteitä missäkin esiintyy. Jännitteitä on yleensä vain kaksi: nolla volttia ja viisi volttia. Kun pisteen jännite on nolla, sanotaan sen olevan loogisessa nollatilassa. Vastaavasti viiden voltin jännite jossakin pisteessä voidaan ilmaista sanomalla kyseisen pisteen olevan loogisessa ykköstilassa.

Tämä jako on vain teoreettinen malli. Käytännön piireissä tulee aina sallia muidenkin jännitteiden olemassaolo, sillä eihän jännitteitä voida edes mitata äärettömän tarkasti. Niinpä onkin sovittu, että loogista nollaa vastaavatkin jännitteet 0 – 0,4 V ja loogista ykköstä jännitteet 2,4–5,0 V.

Tämä määritelmä koskee ns. TTL-standardin piirejä. Jos kytkennästä löytyy näiden alueiden ulkopuolisia jänniteitä, ollaan joko hakoteillä tai jossain toisessa järjestelmässä.

Ohjelmointia harrastanut huomaa pian useita yhtäläisyyksiä näiden ykkösten ja nollien sekä ohjelmointikielten, varsinkin konekielen vastaavien lukujen kanssa. Käytännöllisesti katsoen kaikki koneen sisällä tapahtuva toiminta on vain kahden jännitteen vaihtelua koneen eri mikropiirien välillä. Muistin toimintakin perustuu vain piirin kykyyn pitää yllä sinne syötettyä jännitetasoa.

Binäärikoodi

Tietokoneiden maailmassa joudutaan jatkuvasti tekemisiin sen tosiseikan kanssa, että ihmisten ja tietokoneiden tiedonesitys- ja siirtotavat ovat kovin erilaisia. Koska tietokoneen on tarkoitus olla ihmisen työkalu eikä päinvastoin niin sille pitää "opettaa" jokin ihmismäinen "kieli".

Niinpä tietokoneen valmistaja sitten kehittelee järjestelmän, joka määrää esimerkiksi sen, minkälainen loogisten ykkösten ja nollien yhdistelmä vastaa mitäkin aakkosten kirjainta.

Tilanne muuttuu ongelmalliseksi silloin, kun joku muu kuin koneen valmistaja alkaa tutkia konetta, esimerkiksi rakentaakseen siihen jonkin lisälaitteen. Hänen oman kirjainjärjestelmänsä mukaan tuon koneen toiminnassa ei ole mitään järkeä. Tämän vuoksi tarvitaan yhteisiä järjestelmiä tiedon esittämistä varten.

Sopimuksia tiedon esittämisen periaatteista kutsutaan koodeiksi. Kaikille ohjelmoijille on tuttu ASCII-koodi, joka määrittelee 128 eri kirjainta, numeroa ja muuta symbolia tai toimintoa. Erilaisia koodeja on eri käyttötarkoituksia varten muitakin. Digitaalielektroniikan parissa törmää useimmiten erilaisiin lukujen ja bittien vastaavuuksia määritteleviin koodeihin, joita niitäkin on useita erilaisia.

Tarkastelkaamme nyt kahta tärkeintä digitaalitekniikan koodia. Ensimmäistä niistä kutsutaan nimellä puhdas binäärikoodaus, joka sekin on tuttu useille ohjelmoijille. Binääriluku koostuu bittijonosta eli jonosta ykkösiä ja nollia.

Tällainen jono vastaa jotain ihmiselle tavanomaisemman kymmenjärjestelmän lukua. Binääriluvut puretaan kymmenjärjestelmään seuraavasti:

Kuvitellaan luvun yläpuolelle oikealta vasemmalle kasvava apuasteikko, jossa ensimmäinen luku on 1 ja seuraavat aina kaksinkertaisia edelliseen nähden (1, 2, 4, 8, 16 jne). Valitaan esimerkkiluvuksi vaikkapa 110110:

```
apuasteikko          : 32 16 8  4  2  1
muunnettava luku     : 1  1  0  1  1  0
```

Sitten summataan kaikki ne apuasteikon luvut joiden alla on ykkönen, siis: 32 + 16 + 4 + 2 = 54. Binääriluku 110110 vastaa siis kymmenjärjestelmän lukua 54. Binääriluvun eteen voi, aivan kuten kymmenjärjestelmässäkin, lisätä mielivaltaisesti nollia luvun merkityksen siitä muuttumatta. Niinpä edellinen luku voidaan esittää yhden tavun eli kahdeksan bitin avulla lisäämällä sen eteen kaksi nollaa: 00110110.

Puhtaan binäärikoodin muuttaaminen kymmenjärjestelmään on varsin työläs toimenpide, jos koodin pituus (nollien ja ykkösten lukumäärä) kasvaa kovin suureksi. Sen vuoksi elektroniikassa käytetään yleensä muita koodeja, joista yleisin lienee BCD (eli Binary Coded Decimal = binäärikoodattu desimaali).

BCD-koodi perustuu pitkälti edellä esitettyyn puhtaaseen binäärikoodaukseen. BCD:ssä kymmenjärjestelmän luvun jokainen yksittäinen numero muutetaan neljän bitin puhtaaksi binääriluvuksi. Luku 154 olisi siten BCD:llä ilmaistuna 0001 0101 0100 (0001 => 1 , 0101 => 5 ja 0100 => 4), kun se puhtaalla binäärikoodilla olisi 10011010. Taulukossa A on esitetty bittiyhdistelmien 00000–10011 tulkinta sekä puhtaana binäärikoodina että BCD-koodina.

Binääri	Desimaali	BCD	Binääri	Desimaali	BCD
0000	0	0	1010	10	–
0001	1	1	1011	11	–
0010	2	2	1100	12	–
0011	3	3	1101	13	–
0100	4	4	1110	14	–
0101	5	5	1111	15	–
0110	6	6	10000	16	10
0111	7	7	10001	17	11
1000	8	8	10010	18	12
1001	9	9	10011	19	13

Taulukko A. Binäärilukujen 00000–10011 tulkinta kymmenjärjestelmään puhtaana binäärikoodina ja BCD-koodina. Viiva tarkoittaa sitä, että kyseiselle bittiyhdistelmälle ei ole määritelty arvoa (kielletty arvo).

Yksittäisen kymmenjärjestelmän numeron (0–9) esittämiseen tarvitaan BCD-koodissa neljä bittiä. Neljällä bitillä olisi kuitenkin mahdollista esittää 16 erilaista lukua (2 ^ 4 = 16), joten BCD-koodi tuhlaa osan käytettävissä olevasta muistista. Hukkaan menevän muistin osuus vieläpä kasvaa luvun pituuden kasvaessa. Kahdeksalla bitillä voidaan BCD-koodattuna esittää luvut 0–99, kun ohjelmoija yleensä automaattisesti yhdistää kahdeksan bittiä lukuihin 0–255.

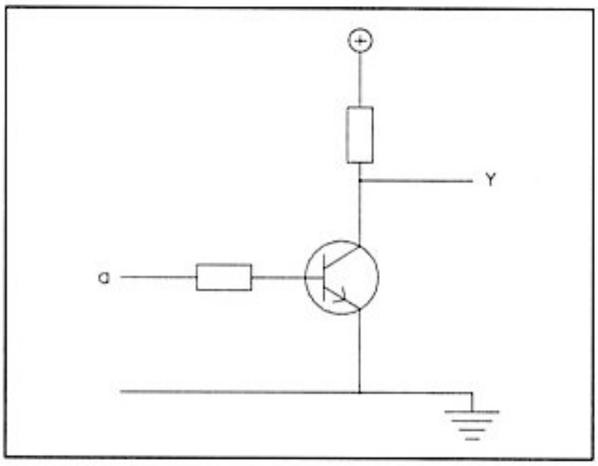
BCD-koodin heikkoudesta seuraa kuitenkin eräs lyömätön ominaisuus. Yhteen kymmenjärjestelmän lukuun vaikuttavat ainoastaan nuo neljä bittiä. Vaikka kone käsittelisi tuhatnumeroisia lukuja, niin yhtä tuon luvun numeroista määriteltäessä tarvitsee tutkia vain neljää bittiä. Tästä seuraa se, että numeroita käsittelevistä koneista tulee huomattavasti selväpiirteisempiä BCD-koodia käyttämällä kuin puhtaalla binäärikoodilla.

Mielenkiintoisena sivutietona mainittakoon, että BCD-koodia käytetään myös joidenkin ohjelmointikielten matemaattisissa toiminnoissa silloin, kun halutaan päästä eroon muuten niin tavallisista koneen pyöristysvirheistä. Tällainen kieli on esimerkiksi vanhan Turbo-Pascal 3.0:n BCD:tä tukeva versio TurboBCD.

Perusportit

Tarkastellaanpa aluksi kuvassa 5.1 esitettyä yksinkertaista yhden transistorin kytkentää. Samantapainen kytkentä esitettiin aiemmin transistorin toimintaa tutkittaessa. Erona on vain se, että kollektoriin on nyt liitetty ylimääräinen johdin, jota merkitsemme Y-kirjaimella.

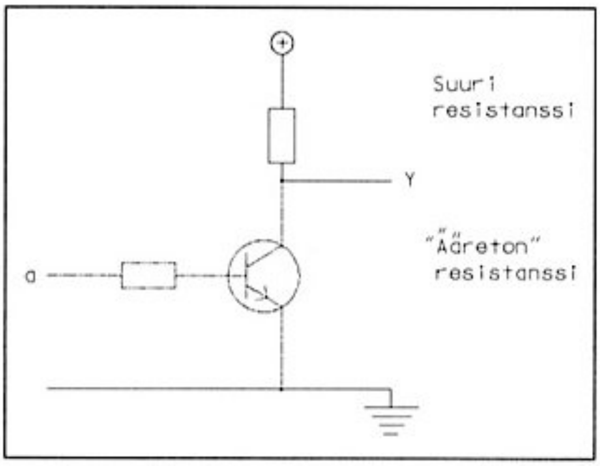
Kuten aiemmin opimme, transistori ei johda virtaa, jos sen kannan jännite on nolla volttia (eli jos se on loogisessa nollatilassa). Olettakaamme siis, että kuvan 5.1 pisteessä a ei ole jännitettä. Tällöin ei myöskään transistorin kannalla ole jännitettä, joten transistori ei johda; sen resistanssi on "ääretön".



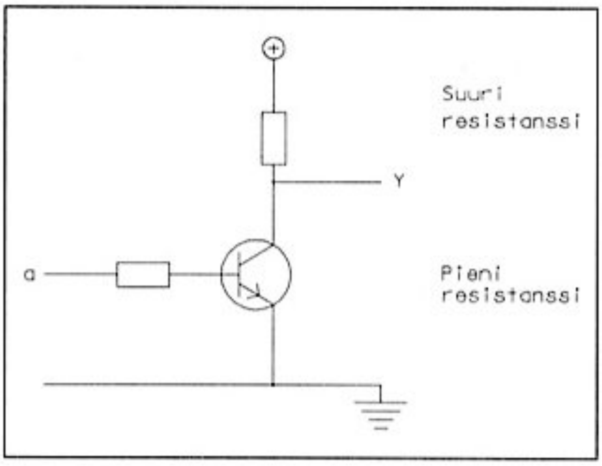
Kuva 5.1. EI-portti.

Kytkentä käyttäytyy kuvan 5.2 mukaisesti ikäänkuin koko transistoria ei olisi olemassakaan. Koska piste Y on nyt kytketty ainoastaan vastuksen kautta plussaan, on sen jännite likimain sama kuin plussan, eli viisi volttia (looginen ykkönen).

Entäpä sitten, kun pisteessä a on viiden voltin jännite (eli se on loogisessa ykköstilassa)? Tällöin transistorin kannalla on positiivinen jännite ja transistori siis johtaa. Jos nyt olemme kaukonäköisesti valinneet plussan ja kollektorin välisen vastuksen riittävän suureksi, on johtavan transistorin sisäinen vastus huomattavasti tätä pienempi, joten transistori "vetää" pisteen Y maahan eli jännitteeseen nolla volttia (looginen nolla) (kuva 5.3).



Kuva 5.2. EI-portti kääntää nollan



Kuva 5.3. EI-portti kääntää ykkösen

Taulukko 5.1			**Taulukko 5.2**			**Taulukko 5.3**		
EI			TAI-EI			TAI		
a	Y		a	b	Y	a	b	Y
0	1		0	0	1	0	0	0
1	0		0	1	0	0	1	1
			1	0	0	1	0	1
			1	1	0	1	1	1

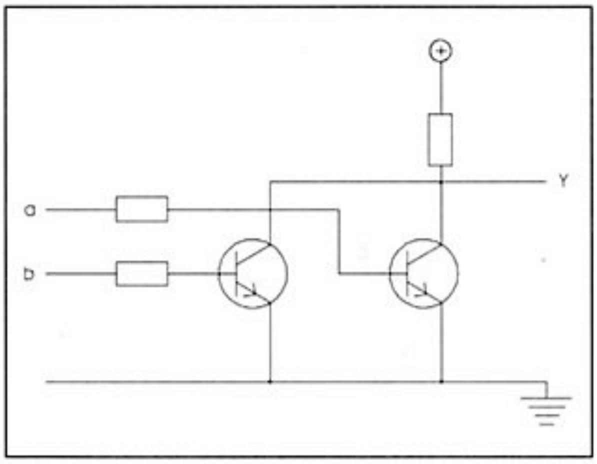
EI-portin, TAI-EI-portin ja TAI-portin totuustaulut.

Tämä piiri toimii siis seuraavasti: Piste Y on nollatilassa silloin, kun piste a on ykköstilassa. Toisaalta kun piste a on nollatilassa, on piste Y ykköstilassa. Piirin piste Y siis antaa pisteestä a ottamansa tilan käänteisenä. Kutsukaamme siis pistettä a ottopisteeksi ja pistettä Y antopisteeksi, tai lyhyemmin vain otoksi ja annoksi.

Piirin toimintaa voi vielä havainnollistaa taulukolla 5.1. Siinä on ylärivillä vierekkäin otto- ja antopisteiden symbolit a ja Y. Symbolin a kohdalle on allekkain merkitty kaikki mahdolliset kyseisen pisteen loogiset tilat, joita tässä tapauksessa on siis vain kaksi (nolla ja yksi). Kullekin vaakariville symbolin Y kohdalle on sitten merkitty kyseisellä rivillä olevan oton tilaa vastaava annon tila.

Tästä taulukosta on helppo etsiä kutakin oton tilaa vastaava annon tila. Tällainen taulukkoesitys saattaa tuntua turhalta hienostelulta, mutta kuten myöhemmin huomaamme, on vastaava menetelmä kätevä sitten, kun ottojen ja antojen määrät kasvavat suuremmiksi.

Lisätkäämme nyt kuvan 5.1 kytkentään toinen transistori siten, että saamme kuvan 5.4 mukaisen kytkennän. Lisätyn transistorin kantaa ohjattakoon vastuksen kautta pisteestä b. Niin kauan kuin piste b on nollatilassa, ei uusi transistori johda, joten kytkentä toimii aivan kuten kuvassa 5.1 oleva. Jos taas piste b on ykköstilassa, vetää uusi transistori annon Y nollatilaan aivan kuten kuvassa 5.3. Koska piste on nyt vedetty nollatilaan, ei pisteen a tilalla ole mitään

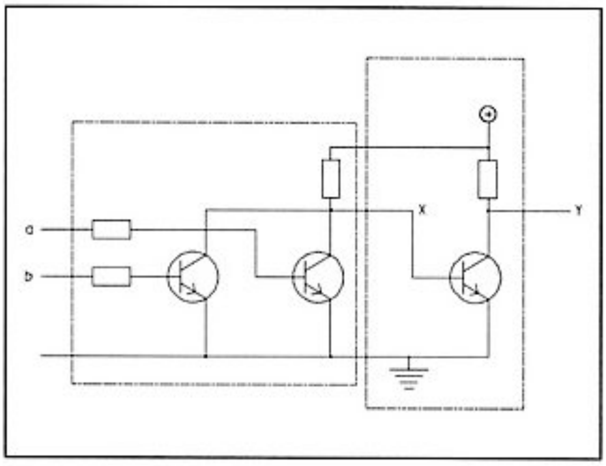


Kuva 5.4. TAI-EI-portti.

vaikutusta annon Y tilaan. Samoin jos piste a on ykköstilassa ja vanha transistori siis vetää annon Y nollatilaan, ei pisteen b tilalla ole vaikutusta antoon Y.

Kuvan 5.4 kytkennän toimintaa voi siis kuvata seuraavasti:

Jos kumpi tahansa tai molemmat otoista (a tai b) ovat ykköstilassa, on anto Y nollatilassa. Kytkennän toiminta voidaan jälleen esittää taulukon avulla (taulukko 5.2). Koska ottoja on nyt kaksi (a ja b), on ottojen tiloja yhteensä neljä kappaletta.



Kuva 5.5. TAI-portti.

Jos kytkemme kuvan 5.4 mukaisen piirin antoon kuvan 5.1 mukaisen piirin oton ja käsittelemme sitä yhtenä kokonaisuutena, saamme kuvan 5.5 mukaisen otuksen. Jos nyt a TAI b on ykköstilassa, on piste X nollatilassa, koska piirin katkoviivalla erotettu vasen puoli on täsmälleen kuvan 5.4 kaltainen. Koska oikea puoli taas on kuvan 5.1 kaltainen, saadaan pisteestä Y pisteen X tila käänteisenä.

Mitä tästä sitten hyödyttiin? Jos kuvan 5.5 piirin toimintaa tarkastellaan taulukon avulla (taulukko 5.3), huomataan että anto Y on ykköstilassa silloin, kun otto a TAI otto b on ykköstilassa. Niinpä kutsummekin tätä kytkentää TAI-piiriksi. Kuvan 5.1 kytkentää taas kutsutaan sen kääntävän toiminnan vuoksi yksinkertaisesti kääntäjäksi tai EI-piiriksi.

Edellä esitetyt kaksi piiriä ovat digitaalielektroniikan peruspiirejä, joita yleensä kutsutaan porteiksi. Portteja on toki muitakin kuin EI ja TAI. Kaikki ne kuitenkin rakentuvat suhteellisen yksinkertaisista transistorikytkennöistä.

Kolme tavallisinta porttia kantavat nimiä JA, TAI ja EI. Koska TAI-portti edellä esitetyn kuvauksen mukaisesti antaa ykkösen silloin kun otto a TAI otto b on ykkönen, ei ole vaikeaa arvata JA-portin toimintaa. Se antaa luonnollisesti ykkösen silloin, kun otto a JA otto b ovat ykkösiä.

Tavallisesti porttien toimintaa kuvataan samanlaisilla taulukoilla kuin edellä TAI- ja EI-portin yhteydessä. Näitä taulukoita kutsutaan virallisesti totuustauluikoiksi. Taulukkoon 5.4 on kerätty kaikkien kolmen peruspiirin totuustaulukot. Totuustaulukon tarkoitus on siis kertoa yksikäsitteisesti, mikä portin annon looginen tila seuraa kustakin ottojen yhdistelmästä. Jos esim. JA-portin ottoihin tuodaan signaalit nolla ja yksi, nähdään taulukon 5.4 JA-portin kohdalta, että ottojen tilaa a=0,b=1 vastaa annon tila Y=0, joten annossa on looginen nolla.

JA			TAI			EI	
a	b	Y	a	b	Y	a	Y
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

Taulukko 5.4. Perusporttien totuustaulukot. Pienet kirjaimet a ja b ovat ottojen ja iso Y annon nimiä.

EHDOTON-TAI		
a	b	Y
0	0	0
0	1	1
1	0	1
1	1	0

Taulukko 5.5. EHDOTON-TAI -portin totuustaulu. Kirjainsymbolit kuten taulukossa 5.4.

TAI- ja JA-porteissa voi olla myös useampia kuin kaksi ottoa. Tällöin totuustaulukko tulee laajemmaksi, mutta toiminnat ovat aina samat:

JA-portin anto on ykkönen vain silloin, kun kaikki otot ovat ykkösiä. Jos yksikin otto on nolla, antokin on nolla.

TAI-portin anto on ykkönen jos yksikin otto on ykkönen. Anto on nolla vain, jos kaikki otot ovat nollia.

On olemassa vielä eräs hiukan harvinaisempi porttityyppi, jota kutsutaan nimellä EHDOTON-TAI. Se poikkeaa tavallisesta TAI-portista siten, että sen anto on ykkönen silloin, kun jompikumpi otoista, mutta ei molemmat, ovat ykkösiä. Siis:

EHDOTON-TAI-portin anto on ykkönen silloin, kun otot ovat erisuuria.

EHDOTON-TAI-portin totuustaulu on taulukossa 5.5.

Tarkastelkaamme vielä kuvaa 5.5. Katkoviivalla rajattu oikea puolisko ei tehnyt mitään muuta kuin käänsi pisteen X tilan päinvastaiseksi. Varsinainen ottojen a ja b "tutkiminen" tapahtuu siis piirin vasemmassa puoliskossa ja tästä saatu "raaka" tulos on pisteessä X.

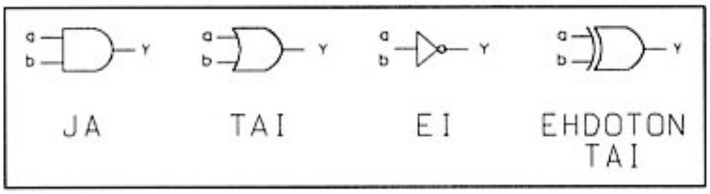
Pisteestä X saatava tulos on vain päinvastainen siihen nähden, mitä sana "tai" tavallisesti tuo mieleen. Vasen puolisko on kuitenkin rakenteeltaan huomattavasti yksinkertaisempi kuin koko kuvassa 5.5 esitetty kytkentä. Niinpä varsinkin laajoissa kytkennöissä olisi paljon taloudellisempaa käyttää vain näitä "vasempia puoliskoja", koska se yksinäänkin osaa suorittaa eräänlaisen TAI-toiminnon.

Käytännössä nämä käänteiset portit ovatkin niiden yksinkertaisuuden vuoksi jopa yleisempiä kuin "oikeat" portit. Porttien nimet ovat tällöin JA-EI, TAI-EI ja EHDOTON-TAI-EI. Nimet johtuvat siitä, että porttien voidaan ajatella rakentuvan emoportista ja kääntäjästä eli EI-portista. Esimerkiksi JA-EI-portti voidaan rakentaa kytkemällä JA-portin antoon EI-portin otto ja käyttämällä sa-

JA-EI			**TAI-EI**			**EHDOTON-TAI-EI**		
a	b	Y	a	b	Y	a	b	Y
0	0	1	0	0	1	0	0	1
0	1	1	0	1	0	0	1	0
1	0	1	1	0	0	1	0	0
1	1	0	1	1	0	1	1	1

Taulukko 5.6. Käänteisten porttien totuustaulut. Pienet kirjaimet a ja b ovat ottojen ja iso Y annon nimiä.

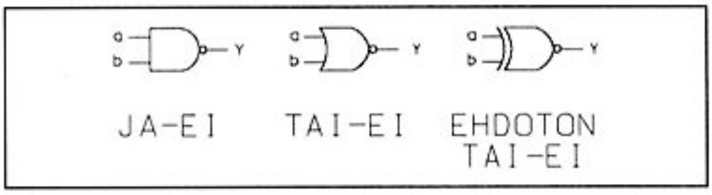
man EI-portin antoa koko yhdistelmän antona. Samoin voidaan JA-portti rakentaa kytkemällä JA-EI-portin anto EI-portin ottoon.



Kuva 5.6. JA-, TAI-, EI- ja EHDOTON-TAI-porttien symbolit.

Käänteisten porttien toimintaa voidaan kuvata sanallisesti näin:

JA-EI-portin anto on nolla vain, kun kaikki otot ovat ykkösiä. TAI-EI-portin anto on nolla, jos yksikin otto on ykkönen. EHDOTON-TAI-EI-portin anto on ykkönen, vain jos otot ovat samoja.



Kuva 5.7. Käänteisten porttien symbolit.

Vastaavat totuustaulukot kaksi ottoa sisältäville porteille ovat taulukossa 5.6.

Koska loogiset portit ovat digitaalielektroniikan "rakennuspalikoita", on niille havainnollisuuden vuoksi määritelty omat piirrosmerkkinsä. Järjestelmiä on käytössä jopa kaksi erilaista, amerikkalainen ja eurooppalainen. Vaikka me asustammekin Euroopassa, käytämme tässä nyt amerikkalaisia piirrosmerkkejä. Syy tähän on se, että käytännöllisesti katsoen kaikki harrastelijan käsiin joutuvat kirjat ja lehdetkin käyttävät niitä. Kuvassa 5.6 ovat JA-, TAI-, EI- ja EHDOTON-TAI -porttien symbolit. Käänteisten porttien symbolit eroavat näistä vain portin antoon piirrettävällä pallukalla, joka kuvaa portin käänteistä toimintaa (kuva 5.7).

Porttien kanssa puuhatessa törmää varsin usein niiden englanninkielisiin nimiin, jotka on siis syytä osata:

JA-portti	**= AND-gate**
TAI-portti	**= IOR-gate**
EI-portti	**= NOT-gate = inverter**
EHDOTON-TAI -portti	**= Exclusive-OR -gate**
	= EXOR-gate = XOR-gate
JA-EI -portti	**= NAND-gate**
TAI-EI -portti	**= NOR-gate**
EHDOTON-TAI-EI-portti	**= Exclusive-NOR -gate =**
	= EXNOR-gate = XNOR-gate

Englanninkielisessä kirjallisuudessa käytetään myös usein numeeristen jännitteiden sijaan kirjaimia L (low) ja H (high). Kirjain L tarkoittaa alempaa (0–0,4 V) ja H ylempää (2,4–5 V) jänniteväliä.

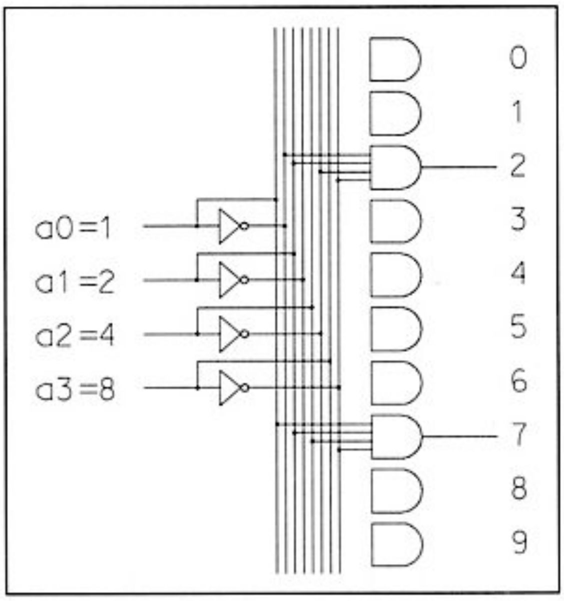
Porttien sovelluksia

Mitä hyödyllistä perusporteilla voisi tehdä? Sovelluskohteita löytyy esimerkiksi aiemmin esitetyistä binäärikoodeista. Oletetaanpa, että meillä on hieno digitaalinen arvontalaite, joka vaikkapa arpoo numeroita nollasta yhdeksään ja antaa tuloksen BCD-koodattuna lukuna. Haluttaessa esittää lukuja koneen käyttäjälle voitaisiin BCD-koodin neljällä bitillä suoraan ohjata esimerkiksi neljää lamppua, jolloin koneen käyttäjä voisi pienellä laskutoimituksella selvittää, mitä kone tahtoo hänelle kertoa.

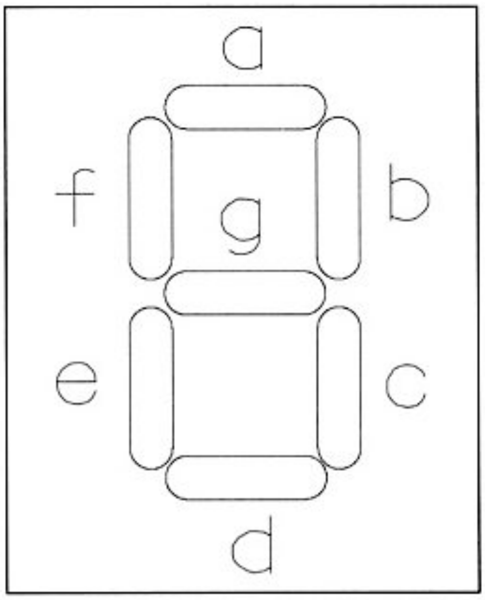
Tällaista esitystä ei voi vielä kutsua kovin ihmisläheiseksi tavaksi. Luku pitäisi esittää siten, että se yhdellä silmäyksellä toisi mieleen tarkoitetun luvun. Parempi tapa olisi käyttää kymmentä lamppua, joista vain yksi kerrallaan palaa ja vastaa siis jotain lukua 0–9.

Rakentakaamme siis sellainen laite, johon syötetään sisään neljä bittiä ja josta tulee ulos kymmenen bittiä, joista vain yksi kerrallaan on loogisessa ykköstilassa vastaten sisään syötettyä binäärilukua.

Juuri tähän voidaan käyttää apuna digitaaliportteja. Tarvittava kytkentä on itseasiassa varsin yksinkertainen (kuva 5.8). Kuvassa on selvyyden vuoksi esitetty vain lukujen 2 ja 7 muuntamiseen vaadittavat langoitukset. Periaatteena koko piirissä on ohjata kymmentä JA-porttia sisääntulevilla signaaleilla ja niiden käänteisarvoilla siten, että kullakin binääriluvun arvolla 0000–1001 (eli 0–9



Kuva 5.8. BCD-yksi kymmenestä -dekooderi.



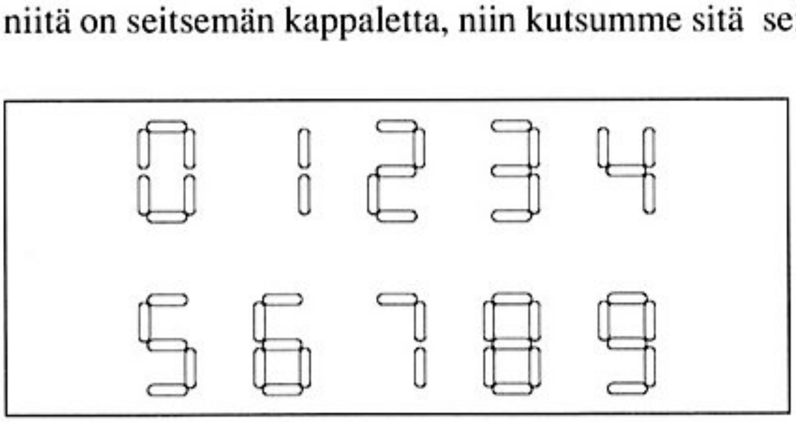
Kuva 5.9. 7-segmenttinäyttö.

kymmenjärjestelmässä) täsmälleen yhden JA-portin kaikki otot ovat ykkösiä, jolloin sen antokin on siis ykkönen.

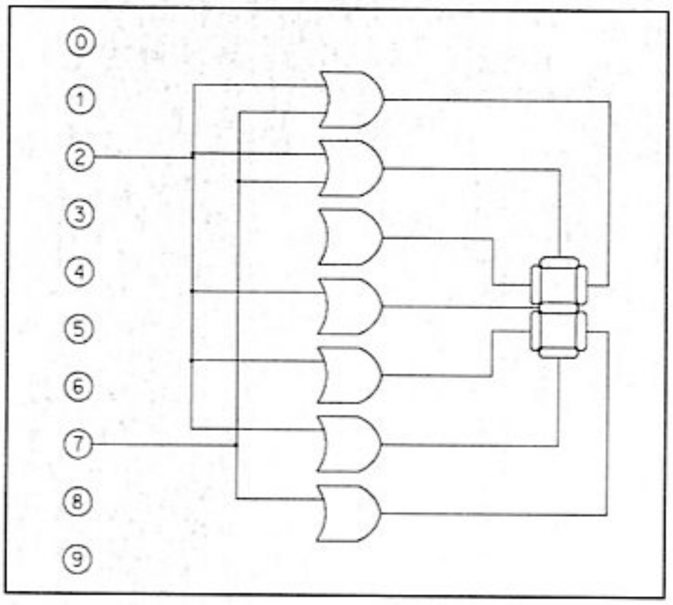
Kytkentä siis valitsee sisään tulevan BCD-koodatun luvun perusteella yhden annon kymmenestä mahdollisesta ykköstilaan ja pitää muut nollatilassa. Piirin nimi onkin "BCD-yksi kymmenestä" -muunnin. Sen voi tietenkin itse rakentaa yksittäisistä porteista, mutta helpommalla ja halvemmalla pääsee hankkimalla valmiin kytkennän sisältävän mikropiirin.

Kymmenen lamppuakaan ei ole välttämättä kaikkein havainnollisin luvun esitystapa. Parempi olisi, jos luvun näkisi suoraan graafisena symbolina. Kuinka numerot sitten voisi esittää vielä havainnollisemmin? Asetetaanpa seitsemän pitkänomaista lamppua erikoiseen asetelmaan kuvan 5.9 mukaisesti.

Näkymä tuo varmasti monelle heti mieleen taskulaskimen tai digitaalikellon. Tällä lamppukuviolla voidaan kätevästi muodostaa numerot 0–9 sytyttämällä tai sammuttamalla lamppuja sopivasti, kuten kuvassa 5.10 on esitetty. Kutakin kuvion lampuista kutsumme hienosti segmentiksi, ja koska näyttöelementissä niitä on seitsemän kappaletta, niin kutsumme sitä seitsensegmenttinäytöksi.



Kuva 5.10. 7-segmenttinäytön lamppuja sytyttämällä voidaan muodostaa numerot.



Kuva 5.11. Yksi kymmenestä-seitsemäksi segmentiksi -muunnin.

Segmentit on nimetty kirjaimilla a–g, kuten kuvassa 5.9 on esitetty. Tätä merkintätapaa käytetään varsin yleisesti alan kirjallisuudessa, joten se on hyvä tuntea.

Koska kullakin segmentillä on vain kaksi tilaa (palaa tai ei pala), voidaan niitäkin oletettavasti ohjata jonkinlaisella digitaaliporttien yhdistelmällä. Koodimuunnoksen suorittava kytkentä on esitetty kuvassa 5.11. Siinä pitää tuoda looginen ykkönen siihen ottoon, jota näytön halutaan esittävän. Muut otot tulee pitää nollatilassa. Otoilla ohjataan seitsemää TAI-porttia, jotka puolestaan ohjaavat näytön segmenttejä. Tässä käytetään TAI-porttia, koska yhden segmentin pitää palaa monilla eri ottojen arvoilla. Esimerkiksi keskimmäinen segmentti palaa, kun otossa on luku kaksi TAI kolme TAI viisi jne. Viivaryteikön välttämiseksi on kuvassa jälleen esitetty langoitukset vain lukujen 2 ja 7 koodausta varten. Piirin nimi on "yksi kymmenestä-seitsemäksi segmentiksi" -muunnin.

Yhdistämällä kuvan 5.8 muuntimen annot tämän piirin ottoihin saamme näppärän piirin, joka muuttaa BCD-koodin luvun suoraan seitsensegmenttinäytön vaatimiksi signaaleiksi. Niinpä yhdistelmäpiirin nimi onkin "BCD-seitsemäksi segmentiksi" -muunnin. Tätäkin muunninta on saatavana valmiina mikropiirinä.

Kiikut (flip-flop)

Tutustukaamme nyt perusporttien jälkeen astetta monimutkaisempiin piireihin, joita kutsutaan kiikuiksi. Ne ovat luonteeltaan varsin erilaisia edellisiin verrattuna, ja niinpä niillä on myös paljon uusia ja mielenkiintoisia sovellutuksia.

Digitaalipiireillä, joihin tähän mennessä olemme tutustuneet, on ollut yhteisenä piirteenä se, että niitä käytettäessä ei ajalla ole ollut juurikaan merkitystä. Antojen tilat ovat riippuneet vain ottojen senhetkisten tilojen yhdistelmistä. Niinpä kutsumme näitä piirejä kombinaatiopiireiksi (combination; yhdistelmä).

Kiikkujen antojen tilat riipuvat paitsi senhetkisistä ottojen tiloista, myös kiikun nk. sisäisestä tilasta. Tarkastelkaamme tilannetta käytännön esimerkillä.

Määritellään uusi ystävämme herra X: Herra X:n tunnetila (sisäinen tila) voi saada vain kaksi arvoa: vihainen tai iloinen. Herra X:n "ottosignaali" olkoon tapahtuma "Herra X tapaa kadulla tuttavansa herra Z:n". "Antosignaali" taas olkoon herra X:n reaktio tuohon tapahtumaan.

Vaikka ottosignaali on kahdella eri tapahtumakerralla täysin sama (X tapaa Z:n), niin X:n sisäisestä tilasta (vihainen tai iloinen) riippuen voi antosignaali (X:n reaktio) olla täysin erilainen, kuten jokainen kokemuksesta tietää.

Palatkaamme sitten taas takaisin elektroniikkaan. Kiikun annon tila riippuu siis sekä ottojen tiloista että kiikun sisäisestä tilasta. Yleensä kiikun sisäisellä tilalla tarkoitetaan sen annon senhetkistä tilaa, joten annon seuraava tila riippuu siis sekä senhetkisistä ottojen että senhetkisestä annon tilasta. Saman päättelyn perusteella voimme taas otaksua, että senhetkisen annon tilan täytyi riippua aiemmista ottojen ja annon tiloista.

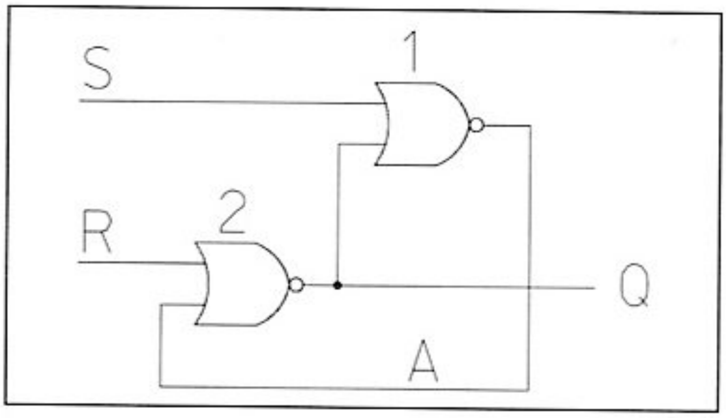
Kiikun toiminta koostuu siis ajallisista jaksoista (seuraava, senhetkinen, aiempi). Niinpä kiikkuja kutsutaankin sekvenssipiireiksi (sequence; jakso, sarja). Sekvenssipiirejä on toki muitakin kuin pelkät kiikut. Käytännöllisesti katsoen kaikki myynnissä olevat valmiit digitaalilaitteet (laskimet, kellot, tietokoneet, jne.) kuuluvat tähän ryhmään.

RS-kiikku

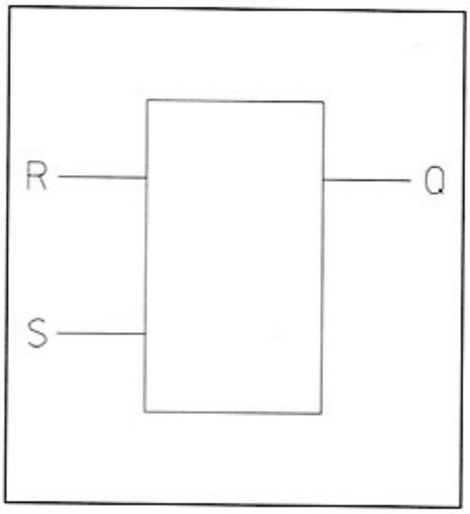
Yksinkertaisin kiikkutyyppi lienee kuvassa 5.12 esitetty RS-kiikku. Se koostuu kahdesta TAI-EI-piiristä, jotka on kytketty hieman oudon näköisesti toisiinsa. RS-kiikulla on kaksi ottoa ja yksi anto. Otot nimetään yleensä tunnuksilla RESET (nollaa) ja SET (aseta), joiden alkukirjaimista koko kiikkukin on saanut nimensä. Normaalisti molemmat otot ovat nollatilassa. Silloin, kun halutaan jotain tapahtuvan, viedään jompikumpi otoista ykköstilaan ja tuodaan se sitten takaisin nollaan.

RS-kiikun toimintaa tutkittaessa pitänee sopia aluksi kiikun annolle jokin alkuarvo, joka olkoon nyt vaikka ykkönen. Koska anto Q on ykkönen ja otto R nolla (otot normaalisti nollassa), täytyy TAI-EI-piirin totuustaulukon perusteella (portti 2) johtimen A olla tilassa nolla. Koska johtimen A tilan taas määrää portti 1, täytyy tuon portin jommankumman tai molempien ottojen olla ykkösiä. Koska Q (portin 1 toinen otto) on ykkönen, tämä ehto täyttyy eikä kytkennässä ole ristiriitaisia tilanteita.

Jos muutamme nyt oton S tilan ykköseksi, portin 1 molemmat otot saavat arvon yksi, mikä ei millään tavoin muuta portin annon eikä siis myöskään piirin minkään muunkaan pisteen tilaa.



Kuva 5.12. RS-kiikku.



Kuva 5.13. RS-kiikun piirrossymboli.

Jos taas otto R saa arvon yksi, muuttuu Q (portin 2 anto) nollaksi. Samalla tietysti portin 1 molemmat otot saavat arvon nolla ja johdin A siten tilan yksi. Taaskaan ei löydy mitään ristiriitaista, vaikka anto Q on vaihtanut tilansa nollaksi. Koska johdin A on nyt ykkönen, ei oton R tilalla ole enää merkitystä; Q pysyy nollana. Muutos saadaan aikaan vain nostamalla otto S jälleen ykköseksi (R:n ollessa nollatilassa), jolloin ympyrä sulkeutuu ja pääsemme takaisin alkutilaan.

RS-kiikun toiminta voidaan siis tiivistää näin:

Annon tila pysyy muuttumattomana, kun molemmat otot ovat nollia. Annon tila muuttuu ykköseksi (tai pysyy ykkösenä), kun otto S käy ykkösessä. Annon tila muuttuu nollaksi (tai pysyy nollana), kun otto R käy ykkösessä.

Jos molemmat otot R ja S asetetaan yhtaikaa ykkösiksi, käskemme kiikkua siirtymään samanaikaisesti sekä ykkös- että nollatilaan. Tämä ei tietenkään ole mahdollista, joten RS-kiikun tilaa ei olekaan määritelty lainkaan tässä tapauksessa. Molemmat otot eivät siis koskaan saa yhtä aikaa olla ykkösiä.

RS-kiikkua ei yleensä piirretä kytkentäkaavioihin kuvan 5.12 kiikkuparina, vaan kuvan 5.13 laatikkona. Samaa periaatetta sovelletaan myös muihin kiikkuihin ja monimutkaisempiinkin digitaalipiireihin. Laatikoina esittäminen yleensä havainnollistaa kytkentöjä ja nopeuttaa niiden piirtoa.

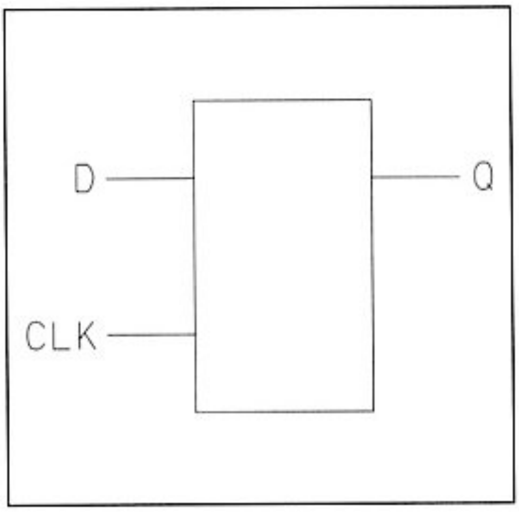
Käyttämällä useampia portteja ja monimutkaisempia kytkentöjä saamme arvatenkin monimutkaisempia ja monipuolisempia kiikkuja. Koska niiden toiminnan selvittäminen porttitasolla veisi huomattavasti tilaa ja koska toisaalta tällainen "ruohonjuuritason" toiminnan ymmärtäminen ei ole kovin tärkeää, tyydym-

me seuraavassa vain tutkimaan muita kiikkuja "mustina laatikoina", joihin viedään jotain ja joista saadaan jotain.

D-kiikku

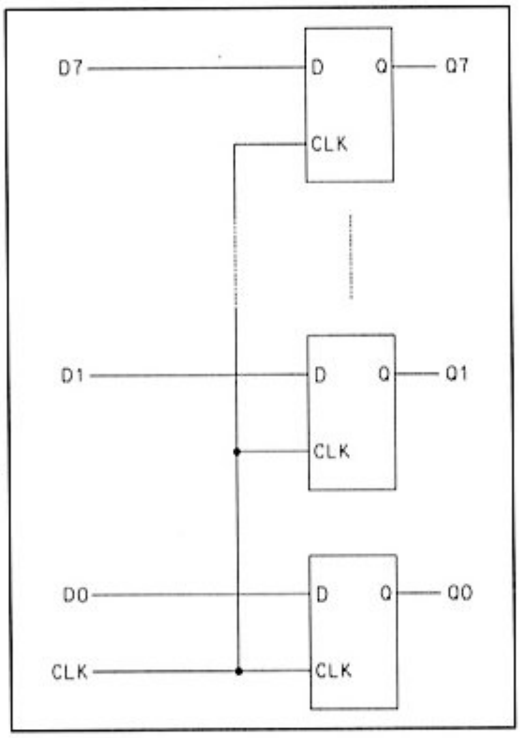
Seuraavaksi tarkastelemme D-kiikkua. Siinä on yksinkertaisimmillaan RS-kiikun tavoin vain kaksi ottoa ja yksi anto (kuva 5.14). Otot ovat nimeltään D eli DATA ja CLK eli kello (clock). Anto lyhennetään jälleen Q:ksi. D-kiikun toimintaperiaate on yksinkertaisuudessaan seuraava:

Kellosignaalin noustessa nollasta ykköseen siirtyy senhetkinen D-oton tila Q-antoon ja pysyy siellä kunnes kellosignaali uudelleen nousee nollasta ykköseen.



Kuva 5.14. D-kiikun piirrossymboli.

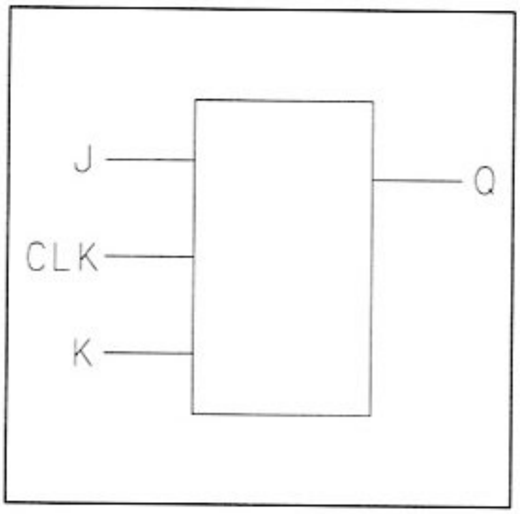
Mitä tämä sitten käytännössä tarkoittaa? Esimerkiksi sitä, että yhtä D-kiikkua voidaan käyttää yhden bitin kokoisena muistielementtinä. Q-annossahan on aina edellisen liipaisuhetken aikainen D-oton tila eli kiikku "muistaa" aiemman tilan. Kytkemällä kahdeksan D-kiikun CLK-otot yhteen saadaan yhden tavun muisti, jota voidaan ohjata yhteisellä kellosignaalilla (kuva 5.15).



Kuva 5.15. Yhden tavun muisti.

JK-kiikku

JK-kiikku (kuva 5.16) on vielä D-kiikkua hieman monimutkaisempi piirielementti. Sillä on peräti kolme ottoa ja yksi anto. Ottojen nimet ovat J, K ja CLK. CLK on jälleen kellosignaali ja annon seuraavan tilan määräävät J ja K. JK-kiikun anto muuttuu D-kiikun tavoin kellopulssin muuttuessa nollasta ykköseen. Annon tila määräytyy sen mukaan mitä



Kuva 5.16. JK-kiikku.

J- ja K-otoissa on tuolla kriittisellä hetkellä. Seuraavat neljä teesiä määrittelevät nuo lainalaisuudet tarkemmin:

Jos J = 0 ja K = 0,
anto pysyy muuttumattomana.

Jos J = 0 ja K = 1,
anto muuttuu nollaksi (tai pysyy nollana).

Jos J = 1 ja K = 0,
anto muuttuu ykköseksi (tai pysyy ykkösenä).

Jos J = 1 ja K = 1,
anto vaihtuu tilansa päinvastaiseksi.

Halutun muutokseen tapahtuminen vaatii siis sekä J:n ja K:n oikeat arvot että kellopulssin muutoksen nollasta ykköseen.

Kiikkujen liipaisutavat

Kellosignaalilla tapahtuvaa kiikun ohjausta sanotaan myös kiikun liipaisuksi. Kiikkua, joka edellä esitetyn mukaisesti liipaistaan kellosignaalin "nollasta ykköseen" -muutoksella, kutsutaan pulssin nousevalla reunalla liipaistavaksi kiikuksi (kello nousee nollasta ykköseen). Jotkut kiikut liipaistaan laskevalla reunalla eli "ykkösestä nollaan" -muutoksella.

Laskevaa reunaa sanotaan myös negatiiviseksi ja nousevaa vastaavasti positiiviseksi reunaksi. Näitä kahta kiikkutyyppiä kutsutaan yhteisesti reunalla liipaistaviksi kiikuiksi.

Hieman harvinaisempi liipaisutapa on pulssilla liipaisu. Pulssilla liipaistavan kiikun kellopulssin pitää muuttua kerran edestakaisin ennen kuin liipaisu on tapahtunut kokonaan. Jos tarvittavat muutokset ovat nollasta ykkösen kautta takaisin nollaan, on kyseessä positiivisella pulssilla liipaistava kiikku. Negatiivisella pulssilla liipaistava tottelee taas kellopulssia ykkösestä nollan kautta ykköseen. Pulssin ajan tulee kiikun muiden ottojen pysyä muuttumattomina.

Pulssilla liipaistavan tyypin hienostuneempi versio on isäntä-renki-tyyppi (master-slave). Tällainen kiikku koostuu itseasiassa kahdesta peräkkäisestä eri reunoilla liipaistavasta kiikusta. Master-slave-kiikun liipaisun jälkeinen tila määräytyy kellopulssin ensimmäisellä reunalla, mutta tuo tila näkyy annossa vasta kellopulssin toisella reunalla.

D-kiikku			JK-kiikku			
D	**Q(n)**	**Q(n+1)**	**J**	**K**	**Q(n)**	**Q(n+1)**
0	0	0	0	0	0	0
0	1	0	0	0	1	1
1	0	1	0	1	0	0
1	1	1	0	1	1	0
			1	0	0	1
			1	0	1	1
			1	1	0	1
			1	1	1	0

Taulukko 5.7. D-kiikun ja JK-kiikun totuustaulukot.

Koska seuraava tila määräytyy ensimmäisellä pulssin reunalla, ei ottojen tarvitse enää sen jälkeen pysyä muuttumattomina. Oleellista kaikissa liipaisutavoissa on se, että niin kauan kuin kello on vakaa, ei annossa tapahdu mitään, vaikka otot muuttuisivatkin.

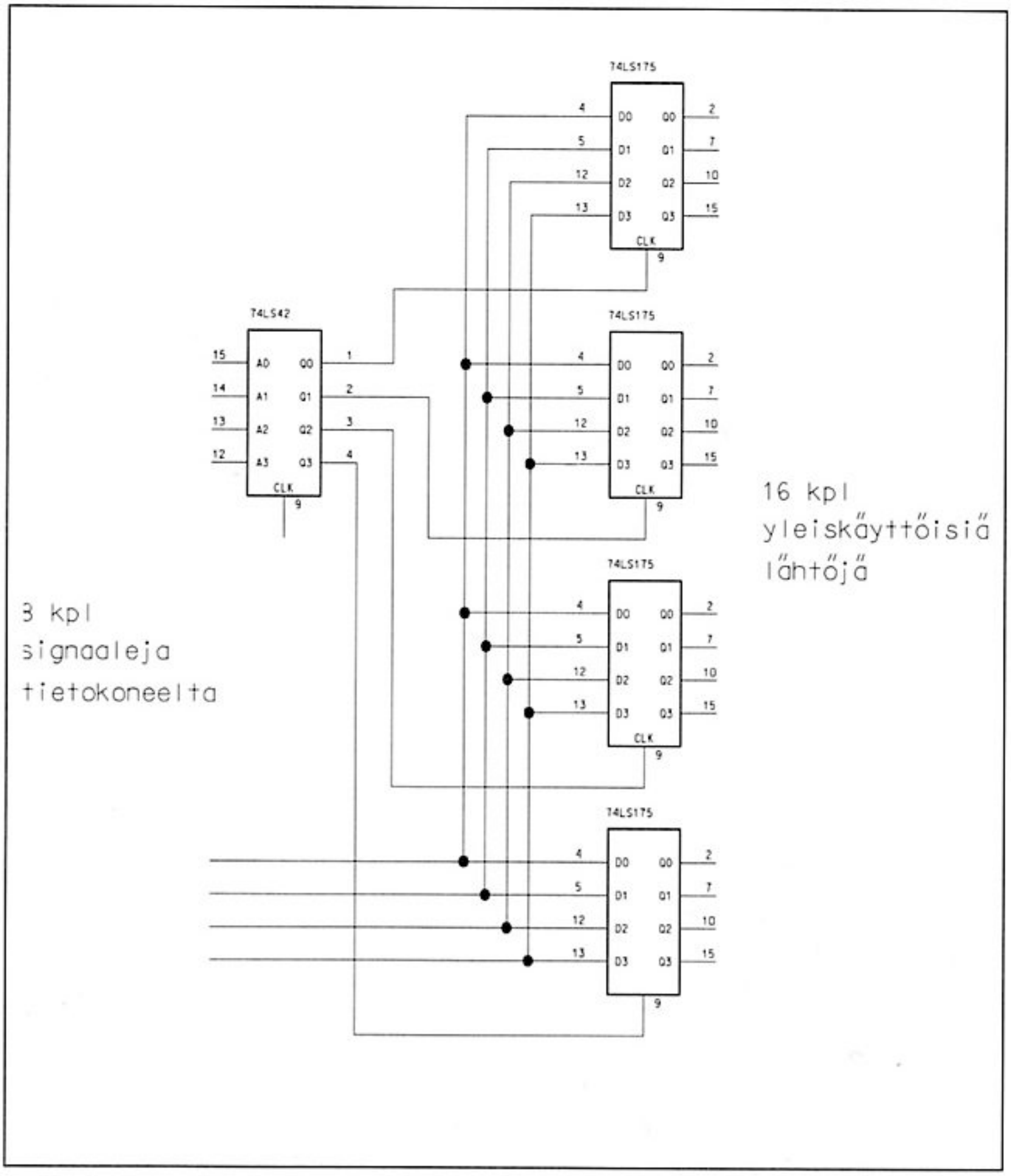
Myös kiikkujen toimintaa voidaan kuvata totuustauluilla. Tällöin tulee taulukkoon yleensä annon Q tilaa kuvaamaan kaksi muuttujaa: Q(n) ja Q(n+1). Näistä ensimmäisellä tarkoitetaan annon tilaa ennen liipaisua ja jälkimmäisellä annon tilaa heti liipaisun jälkeen. D-kiikun ja JK-kiikun totuustaulut ovat taulukossa 5.7.

Kiikkujen sovelluksia

Kuudellatoista D-kiikulla ja yhdellä "BCD-yksi kymmenestä" -muuntimella voimmekin rakentaa mukavan esimerkkikytkennän, joka on vieläpä varsin hyödyllinenkin. Tavallisen kahdeksan bitin käyttäjän portin neljää alinta bittiä käytetään ohjaamaan muunninta, jonka annot puolestaan ohjaavat D-kiikkujen kello-ottoja.

Toiset neljä bittiä kytketään kiikkujen data-ottoihin. Vaikka kytkennässä on kohtalaisen paljon perusyksiköitä, ei valmiissa kytkennässä tarvita kuin viisi mikropiiriä. Näin siksi, että käytetään piiriä 74LS175, joka sisältää 4 samalla positiivisella reunalla liipaistavaa D-kiikkua. Näin ollen mikropiirejä tarvitaan: 4 kpl 74LS175 ja 1 kpl 74LS42.

Piirien yhteishinnaksi tulee noin 50 mk. Kytkentäkaavio on kuvassa 5.17. Tällä laitteella voidaan kaksinkertaistaa käytettävissä olevien portin ohjausnastojen määrä. Laitetta voidaan edelleen laajentaa lisäämällä neljän kiikun piirejä jäljellä oleviin 74LS42:n antoihin. Piirillä 74LS42 on annoissa invertterit, joten syötettyä BCD-koodia vastaava anto on aina alhaalla, kun muut taas ovat ylhäällä.



Kuva 5.17. Mikrotietokoneen käyttäjänportin laajennin.

6. Käytännön digitaalipiirit

Digitaalitekniikan alkuaikoina loogiset portit rakennettiin sähkömekaanisia releitä toisiinsa juottaen. Reletekniikasta on vieläkin häivähdys nähtävillä vanhoissa puhelinkeskuksissa.

Elektroniputki korvasi aikanaan releet nopeampana ja luotettavampana vaihtoehtona. Saman teki myöhemmin myös transistori elektroniputkelle. Edellä transistorein esitellyt yksinkertaiset digitaaliportit ovat sellaisenaan toimivia, mutta nykyään käytettävät digitaalipiirit ovat rakenteiltaan niihin verrattuna huomattavasti monimutkaisempia. Syynä tähän on se, että nuo parilla transistorilla toteutetut portit ovat nykyisiin kytkentöihin liian hitaita ja virtaa kuluttavia.

Erilaisia transistoroitujen digitaalipiirien malleja on aikojen kuluessa kehitetty useita ja ne eroavat toisistaan paitsi sähköisten kytkentöjen, myös valmistusmateriaaliensa osalta.

Samalla materiaalilla ja samoilla kytkentäperiaatteilla toteutettuja piirejä kutsutaan samaan logiikkaperheeseen kuuluviksi. Logiikkaperheitä on nykyään markkinoilla lukuisia, mutta niistä yleisimmät ovat nimiltään TTL ja MOS.

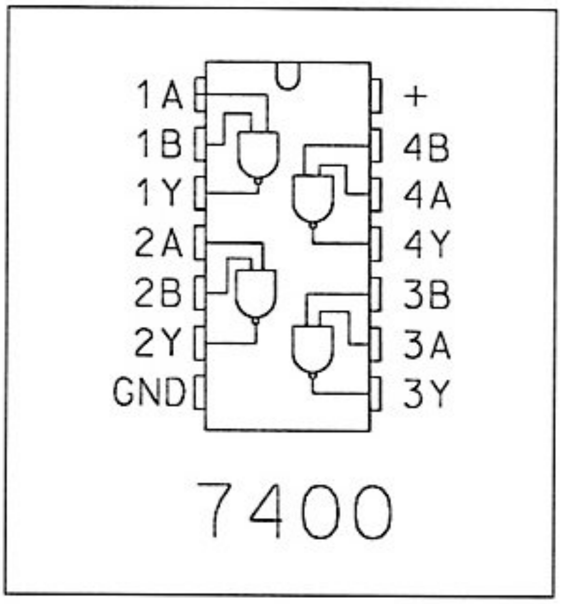
TTL-perhe

Kuten aiemmin mainittiin, vaihtelee jännite digitaalipiireissä yleensä nollan ja viiden voltin välillä. Tämä koskee nimenomaan TTL-perhettä. Lyhenne TTL tulee sanoista transistor-transistor-logic, mikä viittaa siihen, että loogiset toiminnot TTL-perheessä toteuttaa joukko tavallisia transistoreita.

TTL-perheitä on useita hieman erityyppisiä. Nämä poikkeavat toisistaan hieman nopeuden ja virrankulutuksen suhteen. Esimerkiksi Shottky-TTL:nä tai S-TTL:nä tunnettu piiriperhe on hieman perus-TTL:ää nopeampi mutta virrankulutukseltaan suurempi. Muita TTL-perheitä ovat mm. L-TTL (Low power TTL) ja edellisten yhdistelmä LS-TTL. Uudempaa ja nopeampaa TTL-tekniikkaa edustavat AS-TTL- ja ALS-TTL-perheet.

Eräs tunnetuimmista TTL-perheistä on alunperin amerikkalaisen Texas Instruments -yhtiön valmistama niin sanottu 74-sarja.

74-sarjan piirit ovat nykyään yleisiä ja halpoja ja niitä on saatavana kaikkina TTL-perheinä. Esimerkiksi numerojärjestyksessä perheen ensimmäinen piiri



Kuva 6.1. Piiri 7400.

7400 (kuva 6.1) sisältää neljä kappaletta kahden oton JA-EI-portteja. Muiden TTL-perheiden piirit erotetaan perus-TTL:stä lisäämällä nimen kahden ensimmäisen numeron (74) ja varsinaisen piirinumeron väliin kirjaimia. Esimerkiksi 7400-piirin versiot 74L00, 74S00 ja 74LS00 edustavat saman porttirykelmän L-TTL-, S-TTL- ja LS-TTL -versioita.

74-perhe on niin yleinen, että pidemmälle ehtineen digitaalielektroniikan harrastajan kirjahyllystä löytyy miltei aina elektroniikkakaupasta hankittu erillinen perheen satoja piirejä esittelevä kirja, ns. TTL data-book.

TTL-tekniikkaa ei juuri käytetä suuremmissa mikropiireissä suuren virrankulutuksen vuoksi.

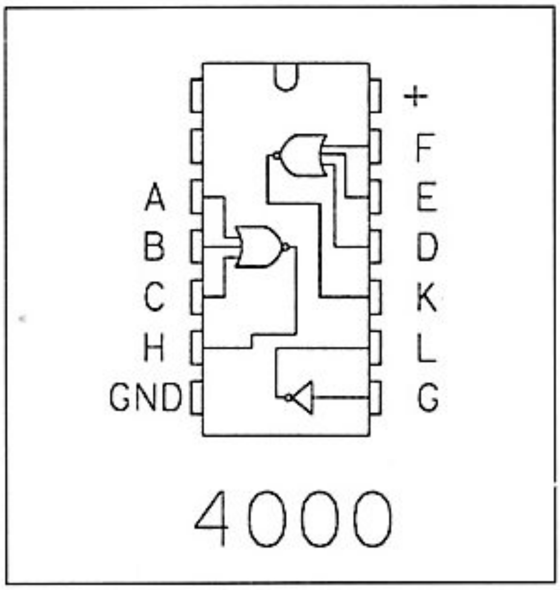
MOS-perheet

MOS-perheitä on oikeastaan useita ja ne kaikki ovat ominaisuuksiltaan huomattavasti TTL:stä poikkeavia. Lyhenne MOS tulee sanoista metal-oxide-semiconductor, eli metallioksidipuolijohde. MOS-piireissä transistorit ovat MOSFET-kanavatransistoreita, jossa transistorin kerrosten väliin jää erittäin hyvin eristävä ohut metallioksidikerros.

Tämä kerros saa aikaan sekä MOS-piirien hyvät että huonot puolet. TTL-piireihin verrattuna MOS kuluttaa vähän virtaa. Toisaalta ohut metallioksidikerros tuhoutuu herkästi staattisen sähkön voimasta.

Puolijohdemateriaalin saostustavan mukaan MOS-piirejä kutsutaan NMOS- tai PMOS-tyyppisiksi aivan kuten tavallisen transistorin osat koostuvat N- ja P-tyypin puolijohteista. NMOS-tekniikkaa käytetään vieläkin melko runsaasti. Esimerkiksi aiemmin suositut prosessorit 6502 ja Z80 edustavat NMOS:ia.

Varsin suosittu logiikkaperhe on myös CMOS, jonka piirit sisältävät sekä NMOS- että PMOS-piirien osaset. CMOS-perheen piirit kuluttuvat virtaa erittäin vähän muihin verrattuna, mutta niiden haittapuolena on herkän vioittumisen lisäksi piirien monimutkainen rakenne. Tästä seuraa väkisinkin se, että tietynkokoiselle piipalalle mahtuu CMOS-logikkaa paljon vähemmän kuin esimerkiksi NMOS-logiikkaa.



Kuva 6.2. Piiri 4000.

CMOS-perheistä yleisin on ns. 4000-sarja. Esimerkiksi sarjan ensimmäinen piiri 4000 (kuva 6.2) sisältää kaksi kappaletta kolmeottoisia TAI-EI-portteja sekä yhden EI-portin.

4000-sarjan eri loogisia tiloja vastaavat jännitteet eivät ole samalla tavoin kiinteät kuin TTL:n. Tämä on hyvin yleistä CMOS-piireille. Loogisten tilojen rajajännitteet riippuvat käyttöjännitteestä, joka puolestaan on vapaasti valittavissa noin 3 ja 18 voltin väliltä.

Loogisten tilojen jännitteet määräytyvät tiettyihin osiin käyttöjännitteestä. Loogista nollaa vastaa 4000-perheessä jännite nollasta voltista aina 30 %:n käyttöjännitteestä. Loogista ykköstä taas vastaa yli 70%:n käyttöjännitteestä oleva jännite. Esimerkiksi 10 voltin käyttöjännitteellä rajat olisivat 0–3 V ja 7–10 V.

Muita logiikkaperheitä

Perusporttien toimintaa esittelevät portit olivat tyypiltään RTL-logiikkaa. RTL (resistor-transistor-logic) on jo jäänyt pois käytöstä, samoin kun sen jälkeläinen DTL (diode-transistor-logic). Sekä RTL:n että DTL:n ovat syrjäyttäneet nopeammat ja paremmin häiriöitä sietävät perheet.

Eräs nopeimmista piiriperheistä on ECL (emitter-coupled-logic). Tätä harvinaista logiikkaperhettä käytetään erittäin nopeissa laitteissa. ECL:n huonoja puolia ovat suuri virrankulutus ja nopeudesta johtuvat piirisuunnitteluvaikeudet.

Saatavana on myös useita muiden perheiden sekoituksia. Näissä pyritään yhdistämään kahden perheen hyvät puolet. Esimerkiksi melko yleinen on 74HC-sarja, jonka piirit ovat muutamin rajoituksin yhteensopivia 74LS-sarjan samannumeroisten piirien kanssa. 74HC-piirit on kuitenkin toteutettu CMOS-tekniikalla, joten niiden virrankulutus on hyvin pieni. 74HC-piirit vastaavat kantakytkennältään samannumeroisia 74LS-piirejä, joten monissa tapauksissa niillä voi suoraan korvata LS-piirit.

	LS-TTL	CMOS	ECL
Käyttöjännite (V)	+5	3 – 18	–5
Tehonkulutus/portti (mW)	2	lähes 0	30
Siirtoviive/portti (ns)	10	30	2

Taulukko 6.1. Logiikkaperheiden vertailua.

Eri logiikkaperheiden erot

Logiikkaperheiden välillä on niin suuria eroja, että ei ole olemassa mitään parasta piiriperhettä, vaan käytettävä perhe tulee aina valita käyttötarkoituksen mukaan. Tätä varten on taulukossa 6.1 likimääräisesti esitelty eri perheiden tärkeimpiä ominaisuuksia.

Kuten taulukosta huomataan, on piirin nopeudella ja tehonkulutuksella selvä suhde: mitä vähemmän tehoa, sitä hitaampi toiminta.

7. Suuret mikropiirit

74- ja 4000-sarjan mikropiirit sisältävät loogisia portteja yleensä muutamasta kappaleesta muutamaan kymmeneen. Nykyisin mikropiireihin pystytään jo ahtamaan jopa satojatuhansia portteja.

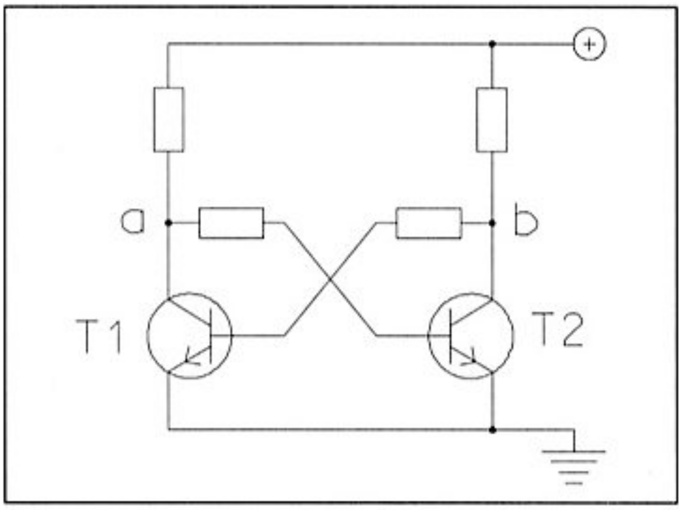
Mikropiirit jaetaan usein kokojensa perusteella luokkiin. Puhutaan SSI- (small scale integration, pieni integrointiaste), MSI- (medium scale integration, keskinkertainen integrointiaste) tai LSI- (large scale integration, suuri integrointiaste) piireistä. Piirien yhä kasvaessa on käyttöön otettu myös LSI:tä suurempia piirejä kuvaavia lyhenteitä, esimerkiksi VLSI (very large scale integration, hyvin suuri integrointiaste).

Tietokoneissa käytetään nykyisin hyvin paljon näitä erittäin tiiviisiin pakkauksiin pakattuja mikropiirejä.

Muistit

Aiemmin totesimme, että D-kiikulla voi varastoida yhden bitin tietoa myöhempää käyttöä varten. Nykyisten mikrotietokoneiden muistit ovat kooltaan jo megatavuja eli miljoonia tavuja. Erillisistä D-kiikuista rakennettu muisti vaatisi suuren määrän mikropiirejä, mikä paisuttaisi laitteen koon jättiläismäiseksi.

D-kiikkuja voisi ahtaa paljon yhdelle piisirulle, jolloin saataisiin sama muistimäärä paljon pienempään tilaan, mutta käytännössä muistipiirit eivät rakennu D-kiikuista vaan huomattavasti yksinkertaisemmista muistielementeistä.



Kuva 7.1. Yhden bitin muistielementti.

Kuva 7.1 esittää erästä yhden bitin muistielementtiä. Siinä on kytketty kaksi tavallista transistoria ristiin. Piirin toimintaa tutkiaksemme valitsemme jonkin sopivan alkutilan. Sovitaan esimerkik-

si, että T1:n kannalla on riittävän suuri jännite avaamaan transistorin johtavaan tilaan. Koska T1 johtaa, vetää se pisteen a maahan (samalla periaatteella kuin aiemmin EI-piirin toimintaa tarkastellessamme). Samalla T1 tietysti vetää myös T2:n kannan maahan, joten T2 ei ole johtavassa tilassa.

Edelleen voimme todeta tästä seuraavan sen, että piste b on vastuksen R2 ansiosta ykköstilassa pitäen samalla yllä T1:n kantajännitettä. Kuvattu tilanne pysyy siis vakaana.

Koska piiri on täysin symmetrinen, on aivan samantekevää kumpi transistoreista johtaa. Toinen on joka tapauksessa aina ei-johtavassa tilassa. Voisimme siis aivan yhtä hyvin tarkastella tilannetta lähtien siitä, että T2 on johtavassa tilassa ja T1 ei. Piiri on vakaa molemmissa asennoissaan, sanomme, että se on bistabiili (bi = kaksi, stabiili = vakaa).

Erikoisia asioita alkaa tapahtua, kun puutumme väkivalloin piirin toimintaan ja pakotamme esimerkiksi pisteen a maahan kytkemällä T1:n kollektorin ja emitterin yhteen. Jos piste a oli ennestään nollatilassa, ei tämä toimenpide luonnollisestikaan mitenkään muuta piirin tilaa. Jos taas piste a oli ykköstilassa, tapahtuukin jotain mielenkiintoista: Piste a ja samalla T2:n kanta menevät nollatilaan, joten T2 sulkeutuu ja piste b ja T1:n kanta nousevat ykköstilaan. T1 alkaa siis johtaa, joten piste a jää nollatilaan, vaikka emme enää pitäisikään sitä "pakko-otteessa".

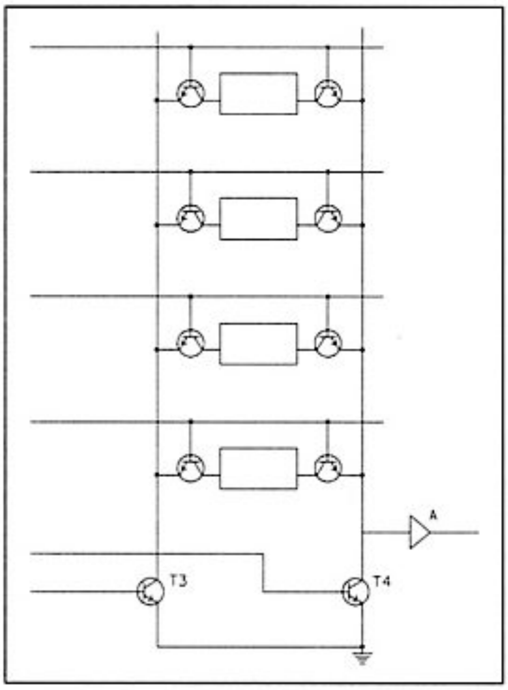
Kytkentä siis vaihtoi tilaansa, kun kytkimme pisteen a hetkeksi maahan. Samalla tavoin päästään takaisin alkutilaan kytkemällä vuorostaan piste b hetkeksi maahan. Koska molemmat tiloista olivat vakaita, on piirillä ilmiselvästi "muistava" ominaisuus. Täytyy vain sopia kumpi pisteistä a ja b määrää muistialkion tilan, sillä niissähän on aina toisiinsa nähden vastakkaiset tilat. Kytkennän toiminta voidaan tiivistää kolmeen muistisääntöön:

Muistielementin transistoreista aina toinen ja vain toinen johtaa kerrallaan.

Muistielementti säilyttää tilansa kunnes se väkisin muutetaan.

Muistielementti voidaan pakottaa kumpaan tahansa tilaan kytkemällä jommankumman transistorin kollektori hetkeksi maahan.

Rakentamalla tuhansia edellä esitetyn kaltaisia muistielementtejä voi rakentaa tuhansien bittien muistin, mutta jos jokaiseen elementtiin kytketään kaksi johdinta muistin lukua ja kirjoitusta varten, syntyy aikamoinen risukasa. Mikropiirin sisällä tällainen johdinrykelmä ei aiheuta päänsärkyä muille kuin piirin valmistajalle, mutta kuinka saada piirikotelon ulkoisten liitäntänastojen määrä pienemmäksi?



Kuva 7.2. Muistipiirin sisäinen rakenne.

Kuvassa 7.2 olevat 4 laatikkoa ovat kukin kuvan 7.1 kaltaisia muistielementtejä. Käyttöjännite- ja maajohtimet on selvyyden vuoksi jätetty piirtämättä. Kunkin elementin luku- ja kirjoituspisteisiin (pisteet a ja b kuvassa 7.1) on kytketty ylimääräiset transistorit, joiden kannat on puolestaan kytketty yhteen vaakasuoralla johtimella. Samoilla pystyriveillä olevien transistorien emitterit on niinikään kytketty yhteen pystysuorilla johtimilla, jotka on edelleen kytketty transistorien kautta maahan.

Jos nyt asetamme yhden vaakasuorista johtimista ykköstilaan ja pidämme muut nollatilassa, avautuvat siihen kytketyt kaksi transistoria, jolloin "pääsemme käsiksi" yhden muistielementin molempiin luku- ja kirjoitusjohtimiin. Kirjoittaaksemme muistiin ykkösen tai nollan tarvitsee meidän nyt vain asettaa jompikumpi pystyjohtimista nollatilaan avaamalla ja sulkemalla transistoreita T3 ja T4.

Jos taasen halutaan lukea jonkin elementin sisältö, asetetaan samalla tavalla yksi vaakasuorista johtimista ykköstilaan ja tunnustellaan pystysuoria johtimia (tai toista niistä). Samalla pidetään molemmat transistorit T3 ja T4 tukossa, jottei muistin sisältö muuttuisi.

Kaikki laitteet, joilla pystyjohtimia muistia luettaessa tunnustellaan, muodostavat resistanssin johtimen ja maan välille, sanomme, että tunnusteleva laite kuormittaa johdinta. Mitä pienempi tuo resistanssi on, sitä enemmän virtaa sen kautta kulkee ja sitä raskaampi on johtimen kuormitus. Jos kuorman resistanssi on riittävän pieni, käy niin, että tunnusteleva laite itseasiassa kytkee pystyjohtimen maahan, mikä edellä todetun mukaan pakottaa muistin toiseen tilaansa (tässä tapauksessa nollatilaan). Jos muisti alunperin oli ykköstilassa, on lukutoiminto aiheuttanut muistin sisällön muuttumisen, mikä ei suinkaan ollut tarkoitus.

Tällaisen tapahtuman välttämiseksi lisätään tunnustelevan laitteen ja johtimen väliin eräänlainen vahvistin (kuvan 7.2 symboli A). Vahvistin nimensä mukaisesti vahvistaa pystyjohtimesta tulevan signaalin sitä liikaa kuormittamatta niin vahvaksi, että sillä voi vaaratta ohjata suurempia kuormia.

Digitaalitekniikassa näitä vahvistimia kutsutaan yleensä puskuripiireiksi. Niitä voidaan käsitellä myös porttipiireinä, joiden totuustaulu on äärimmäisen

yksinkertainen (taulukko 7.1). Puskuripiiri voidaan rakentaa esimerkiksi kytkemällä peräkkäin kaksi EI-piiriä. Huomaa, että puskuripiirin symbolin antoon ei piirretä käänteisyyttä ilmaisevaa ympyrää, kuten EI-piirissä.

otto	anto
0	0
1	1

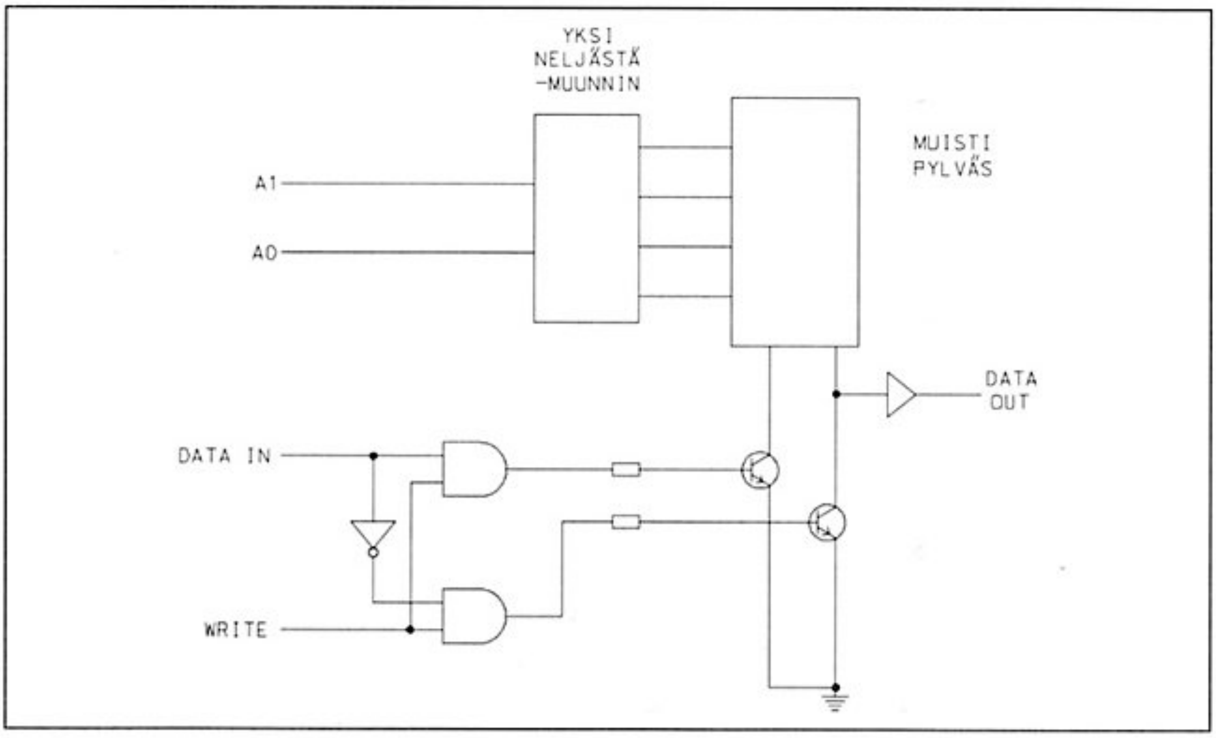
Taulukko 7.1. Puskuripiirin totuustaulu.

Lisäämme vielä joukkoon kuvan 7.3 osoittamat piirit. Kuvassa oleva muistipylväs on kuvan 7.2 kaltainen neljän elementin ryhmä ilman transistoreita T3 ja T4. Kennoston vaakasuoria johtimia, jotka siis valitsevat käsiteltävän muistialkion, ohjataan tavallisella "yksi neljästä"-muuntimella. Tämä varmistaa sen, että vain yksi alkio kerrallaan on valittu ja vähentää samalla tarvittavien vaakajohdinten määrän kahteen (A1 ja A0).

Lisäksi käytämme T3:n ja T4:n ohjailuun kahta JA-piiriä ja yhtä kääntäjää. WRITE-signaalin ollessa alhaalla ei kumpikaan transistoreista saa kantavirtaa, joten molemmat niistä ovat tukossa. Tällöin ottoihin A1 ja A0 tuotu binäääriluku (00–11) valitsee pylväästä yhden muistialkion "avautumaan", jolloin sen sen sisältö voidaan lukea pisteestä DATA-OUT.

Jos taas WRITE-signaali on ylhäällä, voidaan DATA-IN -signaalilla valita jompikumpi transistoreista johtavaksi. Tämä taas saa aikaan A1:llä ja A0:lla valitun muistialkion siirtymisen jompaankumpaan tilaansa.

Oheisessa kytkennässä on portit asetettu siten, että muistialkio mukavasti menee DATA-OUT -pisteestä katsottuna siihen tilaan, missä DATA-IN -piste



Kuva 7.3. Data-, osoite- ja kontrollisignaalien syöttö muistipiirille.

kirjoitushetkellä sattuu olemaan. Koska WRITE-signaalin ollessa ylhäällä aina jompikumpi transistoreista T3 ja T4 on avoimena, tulee WRITE-signaali pitää alhaalla aina muulloin paitsi kirjoitettaessa.

Nyt olemme saaneet rakennettua yksinkertaisen RAM-muistipiirin. Muistin kokoa on helppo kasvattaa korvaamalla "yksi neljästä" -muunnin suuremmalla, jolloin tietysti myös vaakajohdinten (A0, A1...) määrä kasvaa. Käytännön piireissä yksiulotteisen muistipylvään sijasta käytetään yleensä kaksiulotteista matriisia (esimerkiksi 1024x256), jolloin selvitään pienemmillä muuntimilla, mutta joudutaan toisaalta hiukan monimutkaisempiin toteutuksiin. Piirin käyttäjän kannalta muistikennoston muoto on jokseenkin yhdentekevä asia.

Vaakajohtimia (A0, A1...) kutsutaan yleensä muistipiirin osoiteotoiksi. Usein käytetään vielä erillisten DATA-IN- ja DATA-OUT-liitäntöjen asemesta vain yhtä nastaa, joka WRITE-signaalin ollessa ylhäällä toimii DATA-IN-nastana ja vastaavasti DATA-OUT-nastana WRITE-signaalin ollessa alhaalla. WRITE-signaali on käytännössä melko harvinainen, sillä sen tilalla käytetään useimmiten nollatilassaan aktiivista (kirjoitus tapahtuu signaalin ollessa nollatilassa) signaalia.

Tällainen käänteinen toiminto ilmaistaan signaalin nimen päälle vedetyllä viivalla ja signaalin nimi luetaan: "WRITE viiva" ($\overline{\text{WRITE}}$).

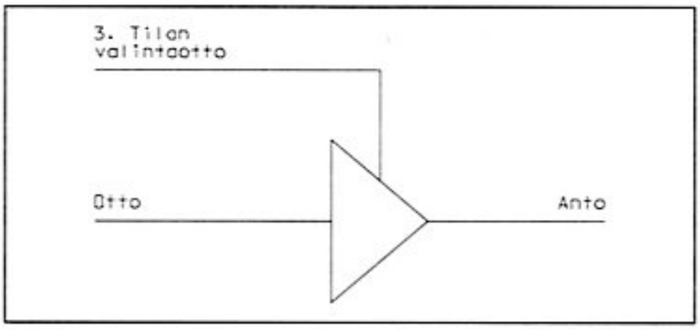
Harrastajan helposti saatavilla on nykyään varsin paljon eri muistikokoja aina yhteen megabittiin asti. Jos valmiina ei kuitenkaan ole saatavan riittävän isoa muistipiiriä, pitää keksiä keino, jolla pienempiä saadaan yhdistettyä.

Yhdistämällä kaikkien muistipiirien osoiteotot ja dataotot/annot keskenään ei vielä päästä kovin pitkälle, sillä pitäisi vielä keksiä jokin tapa valita piireistä yksi kerrallaan käyttöön.

Jos nimittäin kytkemme kahden tavallisen puskuripiirin annot (data-annot) yhteen, syntyy ristiriitainen tilanne, jos toinen niistä sattuu lähettämään ykköstä samaan aikaan kun toinen lähettää nollaa. Tällaisessa tilanteessa saattavat piirit jopa vahingoittua.

Onneksi muistien valmistajat ovat yleensä lisänneet piireihinsä taikanastan nimeltään CS (engl. Chip Select, piirin valinta, tai CE (Chip Enable, piirin sallinta). Tämän nastan toimintaa tarkastellessa törmää jälleen uuteen käsitteeseen nimeltään kolmitila-anto eli TRI-STATE-OUTPUT.

Yleensähän digitaaliportin anto voi saada vain kaksi arvoa, nollan tai ykkösen. Kolmitila-antoinen portti voidaan kuitenkin erillisellä signaalilla asettaa kolmanteen ns. korkeaimpedanssiseen tilaansa, jolloin portti käyttäytyy annosta katsoen aivan kuin sitä ei olisi olemassakaan, tai kuin sen antojohdin olisi katkaistu. Jos siis käytetään puskuripiireinä kolmitila-antoisia tyyppejä, voidaan annot huoletta kytkeä yhteen, kunhan vain varmistetaan, että vain yksi



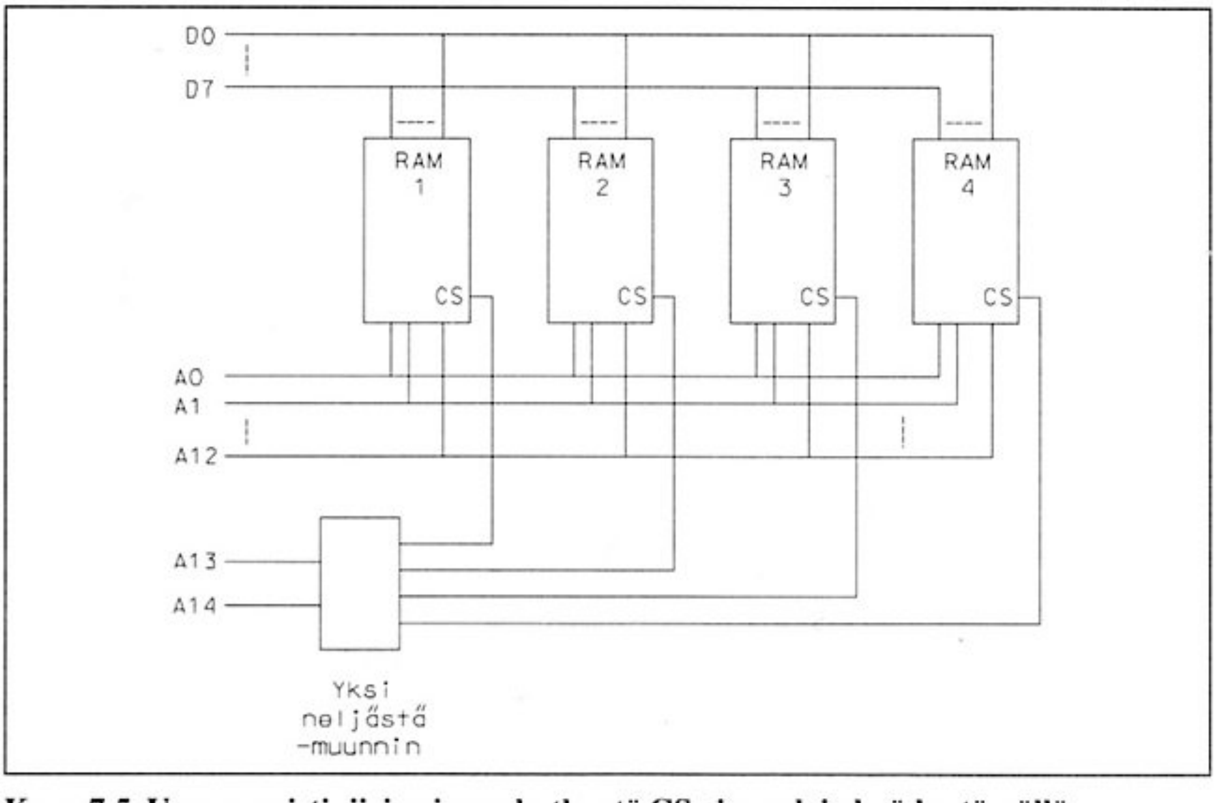
Kuva 7.4. 3-tilapuskuri.

porteista kerrallaan on "hereillä" ja pidetään muut tällöin kolmannessa tilassaan, jolloin niitä ei siis toimivan piirin näkökulmasta katsoen ole olemassakaan.

Kolmitila-anto ilmaistaan yleensä puskuripiirin kylkeen lisätyllä johtimella (kuva 7.4) ja totuustauluissa kolmas tila ilmaistaan merkinnällä Z, High-Z, tms.

Muistipiireissä käytetään yleensä data-antopuskureina juuri tällaisia kolmitila-puskureita, joiden kolmannen tilan valintasignaalit on kytketty yhteen ja nimetty edellä mainituksi CE-signaaliksi. Lisäksi CE:llä ohjataan kirjoitustoimintaa siten, ettei sekään ole mahdollista muulloin kuin CE:n ollessa ylhäällä.

Tällaisia muistipiirejä voi mielinmäärin kytkeä toisiinsa yksinkertaisesti yhdistämällä kunkin piirin data-nastat keskenään, osoitenastat keskenään jne. (kuva 7.5). Ainoastaan CE-signaalit jätetään erilleen, sillä niistä muodostetaan uusia osoite-ottoja. Esimerkiksi neljästä kahdeksan kilotavun muistista voi rakentaa 32 kilon muistin ohjaamalla muistien CE-signaaleja "yksi neljästä" -muuntimella. Samaisen muuntimen valintaotot voi tällöin nimetä uusiksi osoiteotoiksi.



Kuva 7.5. Usean muistipiirin rinnankytkentä CS-signaaleja hyödyntämällä.

Kuvan 7.5 muistipiiri toimii siis siten, että kun osoiteotoissa on osoite väliltä 0–8191 (A13=A14=0), vasemmanpuoleisimman muistipiirin CE-otto on ylhäällä, jolloin luku ja kirjoitutoiminnot koskevat siis vain sitä muiden ollessa ainoastaan "kuulolla". Osoitteilla 8192–16383 käytetään toista piiriä jne.

Muut erikoispiirit

Mikrotietokoneissa käytetään paljon erilaisia liityntäpiirejä, jotka ovat rakenteeltaan usein tavanomaisia porttikytkentöjä huomattavasti monimutkaisempia. Esimerkiksi modeemin kytkeminen vaatii mikrolta kyvyn lähettää tietoa sarjamuodossa yhtä johdinta pitkin yksi bitti kerrallaan. Tätä varten valmistetaan erilaisia sarjaliityntäpiirejä, jotka on yleensä helppo liittää piirin valmistajan prosessoreihin.

Sarjaliityntäpiiri huolehtii siitä, että kukin bitti siirtyy ajallaan siirtoväylälle. Samalla piiri voi myös tarkistaa esimerkiksi vastaanotettujen merkkien oikeellisuutta erilaisten tarkistussummien avulla ja varastoida vastaanotettua tietoa siltä varalta, että itse prosessori ei juuri sillä hetkellä ehdi sitä noutamaan.

Toinen perinteinen liityntäpiiri on rinnakkaisliitynnän toteuttava kytkentä. Esimerkiksi varsin yleinen kirjoittimen ja mikron välinen liitäntästandardi Centronics siirtää tietoa rinnakkaismuodossa yhden tavun kerrallaan. Tätäkin varten on olemassa valmiita rinnakkaisliityntäpiirejä.

Erilaisista liityntäpiireistä saa parhaiten kuvan tutustumalla jonkin mikroprosessorivalmistajan tuotteisiin. Yhden mikroprosessorin oheistuotteiden dokumentit kootaan usein yhteen kirjaan, jota valmistaja sitten myy lähes omakustannushintaan saadakseen käyttäjiä piireilleen.

8. Mikrotietokoneen perusrakenne

Väylät

Johdinrykelmää, jossa kulkee samanaiheista tietoa useiden piiriyksiköiden välillä kutsutaan väyläksi (BUS). Kuvassa 7.5 on kaksikin väylää: käytettävien muistipaikkojen osoitteet kulkevat pitkin osoitelinjojen A0–A14 muodostamaa osoiteväylää ja muistipiirien sisältämä ja niihin kirjoitettava tieto pitkin dataväylää D0–D7. Väylärakenteelle ovat ominaisia seuraavat piirteet:

Vain yksi laite (muistipiiri, prosessori, jne.) kerrallaan voi lähettää tietoa väylää pitkin. Esimerkiksi kuvassa 7.5 vain yksi neljästä muistipiiristä voi kerrallaan lähettää tietoa dataväylälle.

Muut laitteet voivat joko vastaanottaa lähetetyn tiedon tai olla välittämättä siitä.

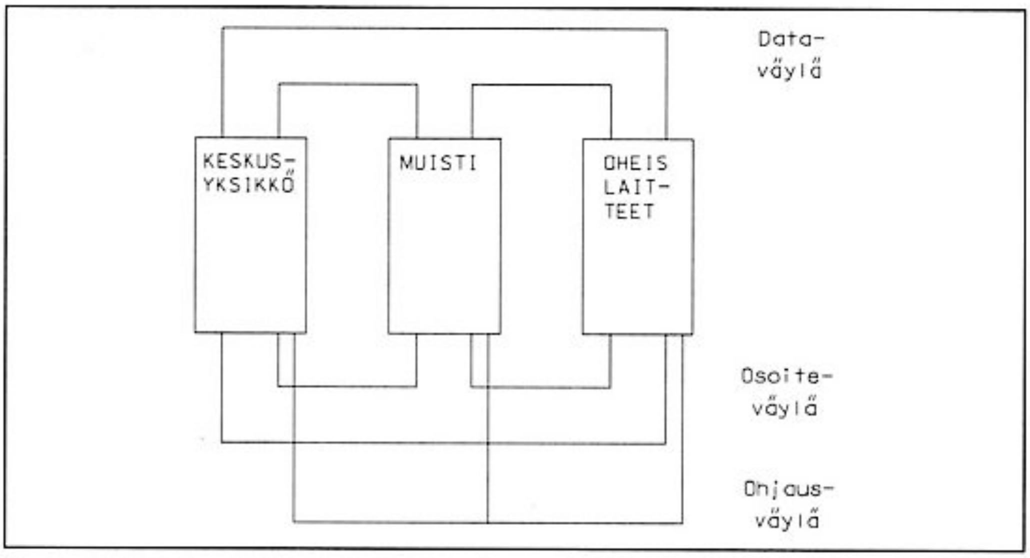
Väylärakenne on oleellinen rakenne kaikissa mikrotietokoneissa. Yleensä väylien "liikennepoliisina" toimii mikron prosessori.

Mikrotietokoneen väylät

Lähes kaiken "älyllisen" toiminnan tietokoneessa suorittaa mikroprosessori. Tästä syystä on varsin luonnollista, että useimmat koneen sisältämät väylät kulkevat prosessorilta muisti- ja oheispiireille. Oheispiirien keskinäiset väylät ovat melko harvinaisia.

Mikrotietokoneiden väyliin pääsee käsiksi laajennusporttien kautta. Koska laajennusporttien nastat on usein kytketty suoraan prosessorin nastoihin, voi niiden kautta kolkutella koneen pyhimpiin osiin johtavia ovia. Tämä tarkoittaa valitettavasti myös sitä, että koneen tuhoaminen omilla "neronleimauksilla" ei ole ainakaan liian vaikeaa.

Vaaraa voi vähentää käyttämällä puskuripiirejä omien laitteiden ja portin nastojen välillä. Useimmiten puskurointi on välttämätöntäkin sen vuoksi, että prosessoreiden liitännät on suunniteltu ohjaamaan ensisijassa oheispiirejä, jot-



Kuva 8.1. Mikrotietokoneen kolme tärkeintä väylää.

ka erityisen valmistusmenetelmänsä (CMOS) ansiosta kuormittavat ohjaajiaan huomattavasti vähemmän kuin TTL-piirit (74-piirit).

Kolme tärkeintä väylää, joita ilman ei nykyisin periaattein toimivaa tietokonetta juuri voisi kuvitella, ovat jo aiemmin esille tulleet osoite- ja dataväylät sekä kolmas, jota kutsutaan ohjaus- tai kontrolliväyläksi (kuva 8.1).

Ohjausväylä kuljettaa piirien välillä sekalaisen määrän tietoja, jotka määräävät erilaisia piirien toimintajärjestykseen liittyviä asioita: esimerkiksi tiedon siitä mihin suuntaan dataväylän tieto kulloinkin on menossa (prosessorilta muistille vai päinvastoin), tai oheispiirien prosessorille lähettämiä "huoltopyyntöjä", ns. keskeytyksiä.

Muistien yhteydessä esitelty WRITE-signaali lienee ohjausväylän signaaleista tärkein. Sen nimi lyhennetään yleensä $\overline{W}$/R:ksi (WRITE/READ). Lisäksi jommankumman, yleensä W:n päälle vedetään viiva osoittamaan tuon toiminnon olevan aktiivinen signaalin ollessa nollatilassa ($\overline{W}$/R).

Jos siis $\overline{W}$/R-signaalin W-kirjaimen päällä on viiva, tarkoittaa se sitä, että prosessori haluaa lukea tietoa (esimerkiksi muistipiiriltä) silloin, kun tuo signaali on ykkönen ja vastaavasti kirjoittaa signaalin ollessa nolla.

Piirivalmistajien käsitykset $\overline{W}$/R-signaalista, kumma kyllä, ovat siinä määrin yhteneviä, että esimerkiksi muistilaajennuksia rakennettaessa voi laajennusportin W/R-signaalin yleensä vetää suoraan muistipiirin vastaavaan nastaan.

Muita tärkeitä ohjausväylän signaaleja ovat erilaiset keskeytyssignaalit. Näiden nimiä ovat esimerkiksi NMI, IRQ ja RES. Näiden suunta on $\overline{W}$/R:stä poiketen oheispiireiltä prosessorille, sillä niiden tarkoituksena on antaa oheispiireille mahdollisuus ohjailla prosessoria.

Keskeytysottojen signaalit ovat yleensä aktiivisia nollatilassa ollessaan, mikä $\overline{\mathrm{W}}$/R:n tapaan ilmaistaan tunnuksen päälle vedetyllä viivalla.

Saadessaan keskeytyssignaalin prosessori keskeyttää suorittamansa ohjelman ja siirtyy toteuttamaan erikseen määriteltyä keskeytystoimintoa. Se mitä tuo toiminto saa aikaan, riippuu käytetystä prosessorista ja keskeytyksestä.

Yleensä keskeytys saa aikaan prosessorin ensin jättämään muistiin erityisen "kirjamerkin", jonka avulla myöhemmin palataan keskeytyneeseen ohjelmaan. Tämän jälkeen luetaan uusi ohjelmaosoite jostain ennaltamäärätystä paikasta.

Luetusta osoitteesta alkavan ohjelman suoritusta jatketaan sitten, kun erityiseen keskeytyksen lopetuskomentoon törmääminen ohjaa prosessorin suorittamaan keskeytynyttä ohjelmaa.

Keskeytystoiminnoilla on helppo järjestää esimerkiksi prosessorin ja lisälaitteiden (näppäimistön, hiiren, jne.) välinen joustava tiedonsiirto.

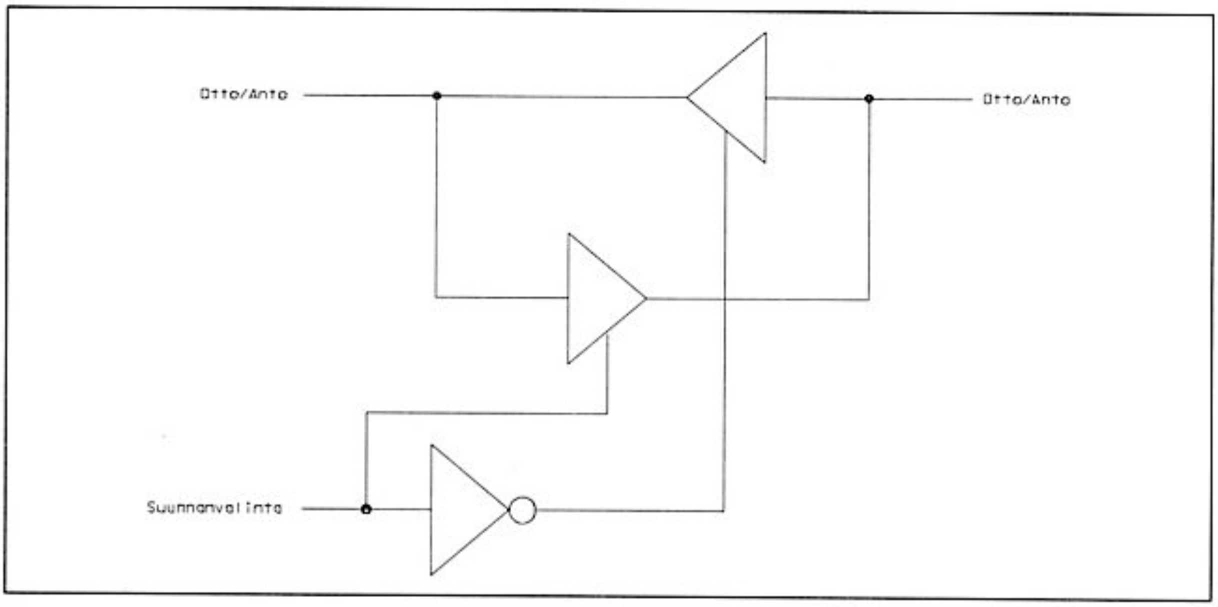
Dataväylän leveys on tavallisesti kotimikroissa 8–32 bittiä. Tätä väylää pitkin kulkevat varsinaiset koneen käsittelemät tiedot: luvut, merkit, konekieliset käskyt jne. Väylän kannalta katsoen ne ovat kaikki vain kaksitasoisten jännitteiden ryhmiä.

Prosessori ja muut piirit kuitenkin tulkitsevat eri aikaan tulleet signaaliryhmät eri tavoin. Esimerkiksi konekielikäskyä seuraa usein jokin käskyä täsmentävä lukuarvo. Niinpä esimerkiksi 6502-prosessorin dataväylältä saamat peräkkäiset luvut 169, 169 ovat sähköisessä mielessä täsmälleen samoja signaaleita. Prosessori tulkitsee kuitenkin niistä ensimmäisen käskyksi LDA # eli "lataa akkuun seuraava dataväylän tuoma luku", joten akkuun (prosessorin eräs sisäinen rekisteri) tallentuu jälkimmäinen 169.

Erikoista dataväylässä on myös se, että siinä tieto kulkee kahteen suuntaan. Prosessorinhan täytyy pystyä sekä lukemaan että kirjoittamaan muistia. Edellisessä tapauksessa muisti toimii lähettäjänä ja prosessori vastaanottajana ja jälkimmäisessä päinvastoin. Väylän suunnan määrää yleensä prosessori ohjausväylän em. $\overline{\mathrm{W}}$/R-signaalilla.

Kaksisuuntaisuuden vuoksi on dataväylän puskurointi hieman muita väyliä monimutkaisempaa (lue: kalliimpaa). Tavalliset puskuripiirit ovat yksisuuntaisia eli niillä on erillinen otto, johon puskuroitava signaali tuodaan ja erillinen anto, josta puskuroitu signaali saadaan.

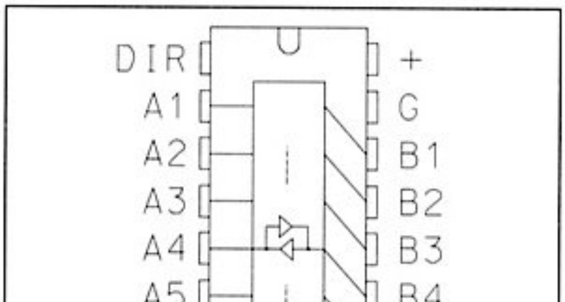
Dataväylän puskurointiin käytetään erikoisia kaksisuuntaisia puskureita (bidirectional buffers), jollaisen periaate selviää kuvasta 8.2. Kaksisuuntainen puskuripiiri koostuu kahdesta vastakkain asetetusta kolmitilapuskurista. Kolmannen tilan asetusotoista toista ohjataan erityisellä suuntasignaalilla ja toista saman signaalin käänteisarvolla (EI-piirin läpi viedyllä signaalilla). Näin ollen ai-



Kuva 8.2. Kaksisuuntaisten puskureiden periaate.

na toinen ja vain toinen puskureista on kerrallaan toiminnassa toisen ollessa vain "kuulolla".

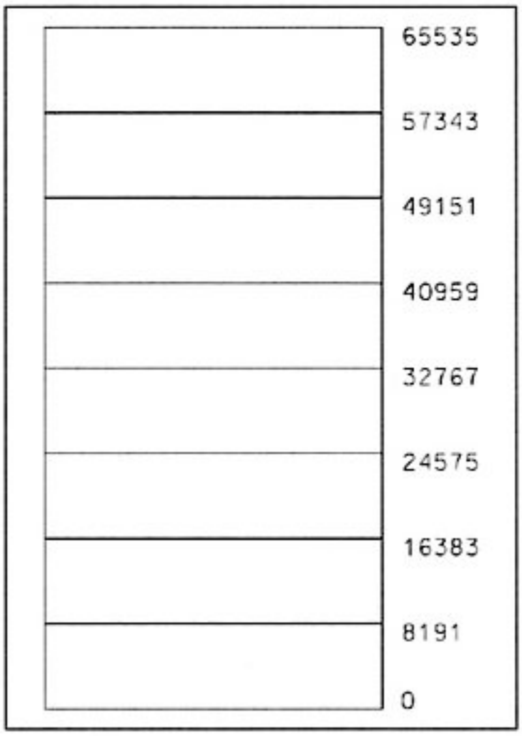
Suuntasignaalina käytetään jälleen prosessorin tarjoamaa $\overline{W}$/R-signaalia. Valmiita kaksisuuntaisia usean bitin levyisiä puskureita voi hankkia esimerkiksi mikropiirinä 74LS645 (8 bittiä). Tässä piirissä on suuntavalintanastan 1 lisäksi nastalla 19 mahdollisuus asettaa kaikki puskurit kolmanteen tilaan, jolloin signaalit eivät kulje kumpaankaan suuntaan. Nastajärjestys on kuvassa 8.3.



Kuva 8.3. Piiri 74LS645.

Osoiteväylä on kotimikroissa useimmiten 16–32 osoitebittiä sisältävä johdinrykelmä. Tämän väylän kautta prosessori määrää kulloinkin käsiteltävän muistipaikan numeron binäärilukuna. Prosessorin lisäksi myös muut piirit lähettävät osoitteita. Esimerkiksi monitorin kuvasignaalin tuottava videopiiri saattaa lukea kuvan muodostamiseen tarvittavat tiedot muistista prosessorin kanssa samoja väyliä käyttäen. Tällaisissa järjestelmissä täytyy tietenkin huolehtia siitä, etteivät piirit koskaan lähetä väylille yhtä aikaa.

Jos osoitebittejä on vanhojen 8-bittisten mikrojen tapaan 16, voidaan niillä osoittaa 65 536 muistitavua (2 ^16 = 65 536). Tuota aluetta (0–65 535) kutsutaan koneen muistiavaruudeksi. Omia

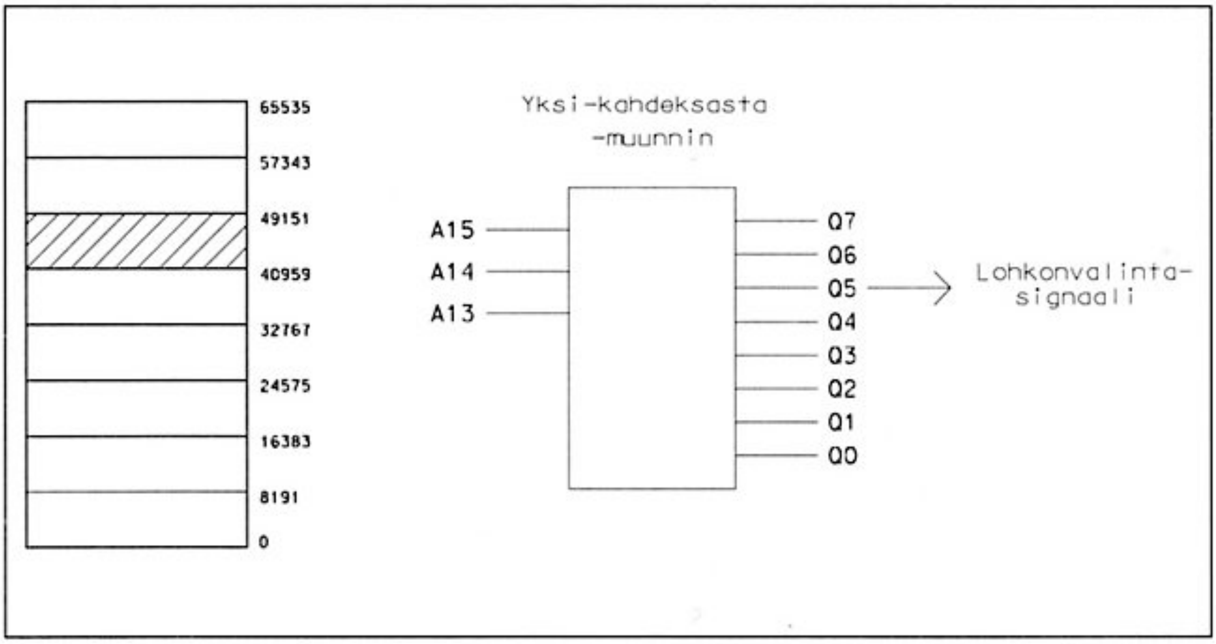


Kuva 8.4a. 64 kilotavun muistiavaruus.

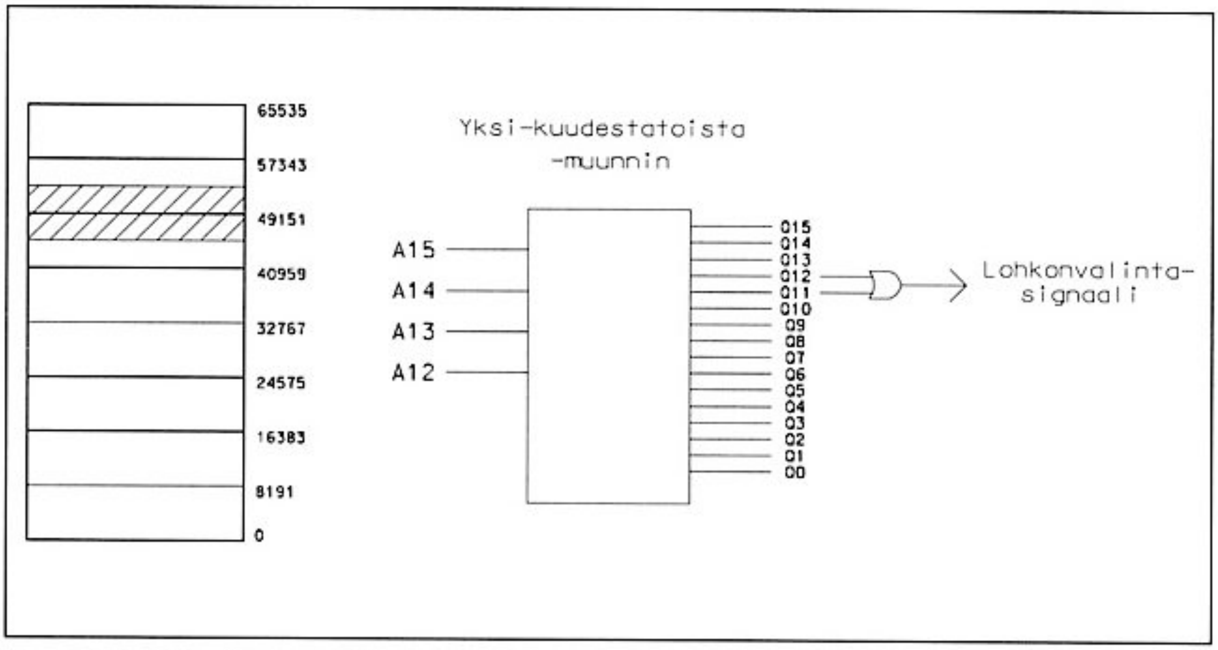
laitteita laajennusporttiin rakennettaessa joudutaan laite aina sijoittamaan sopivaan kohtaan muistiavaruutta, jotta siihen voitaisiin ohjelmassa viitata. Sijoituskohdassa ei saa olla ennestään muita laitteita, joten väylään liitettävän lisälaitteen suunnittelu vaatii melkoista koneen tuntemusta.

Sopivan sijoituspaikan löydyttyä pitää luoda signaali, joka ilmaisee halutulle alueelle kohdistuvat operaatiot. Tämän signaalin tarkoitus on ohjata sekä lisälaitteen toimintaa että dataväylän ja laitteen väliin lisättäviä kolmitilapuskureita, jotta ne eivät lähettäisi samanaikaisesti muiden jo väylällä olevien piirien kanssa. Muistipiirien tapauksessa tällä signaalilla ohjataan tietysti suoraan piirin CS-ottoa.

Valintasignaali muodostetaan osoiteväylän signaaleista joko valmiilla piireillä tai porttipiireistä kootuilla kytkennöillä. Signaalin muodostus on melko helppo tehtävä silloin, kun käytettävä muistialue on jokin kakkosen potenssin (1, 2, 4, 8, 16 jne.) osa koko muistiavaruudesta ja kun se alkaa tällaisen alueen reunasta.



Kuva 8.4b. 8 kilotavun lohkon osoitteiden tunnistaminen yksi-kahdeksasta-muuntimella.



Kuva 8.4c. Limitetyn 8 kilotavun lohkon osoitteiden tunnistus.

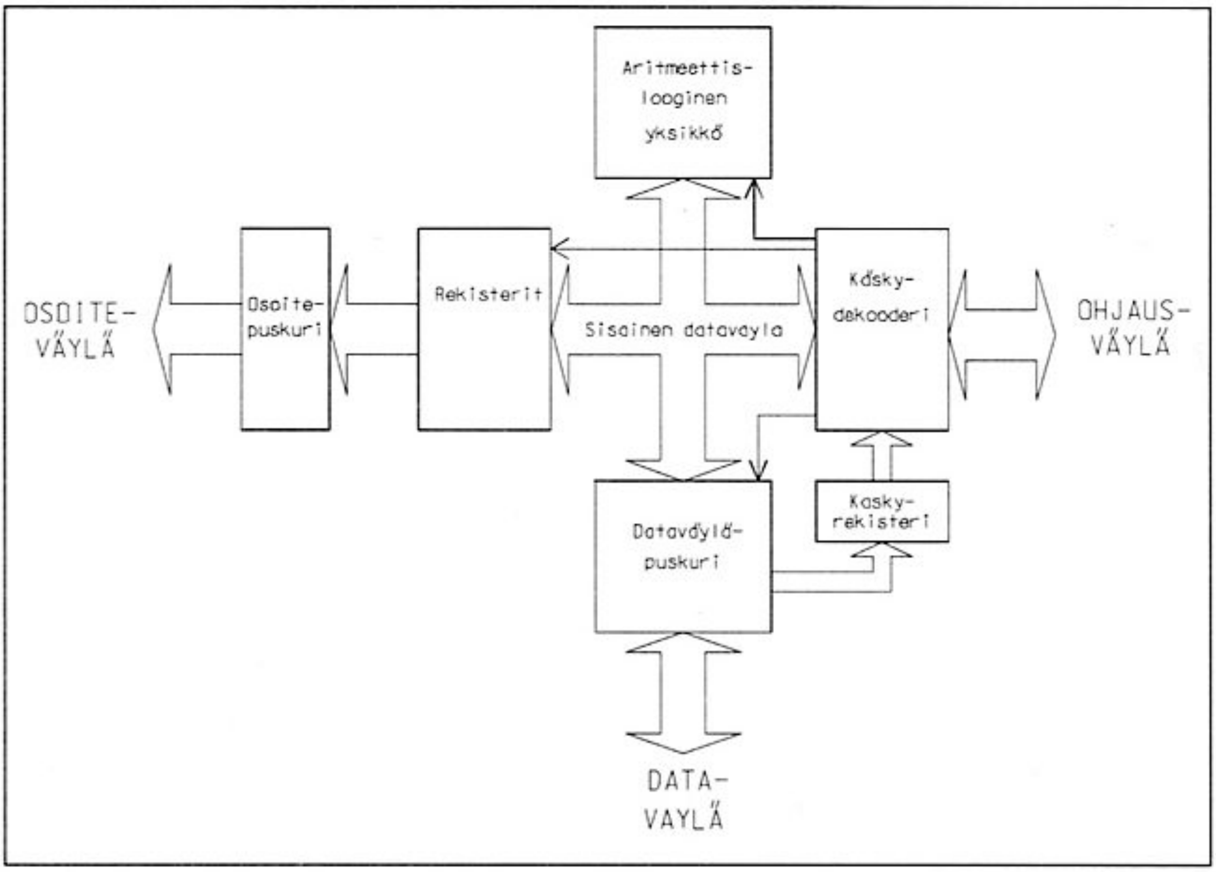
Esimerkiksi 64 kilon muistiavaruuden kahdeksasosa on tasan 8 kiloa. Kuva 8.4a esittää koko muistiavaruutta jaettuna kahdeksan kilon osiin. Ohjaamalla "yksi kahdeksasta" -muunninta osoiteväylän kolmella ylimmällä bitillä saadaan muuntimen annoista aina täsmälleen yksi ykkönen ja seitsemän nollaa. Ykkösenä on aina se signaali jonka numero on sama kuin sen muistiavaruuden lohkon järjestysnumero, jota väylä parhaillaan osoittaa. Näin ollen minkä tahansa kahdeksan kilon lohkon, joka sijaitsee sopivasti jossain kuvan 8. 4a "lokeroista", valintasignaali saadaan suoraan jostain muuntimen annosta (kuva 8.4b).

Tilanne muuttuu hankalammaksi, jos haluttu muistialue sijaitsee limittäin kahden lokeron alueella. Mitä epäsäännöllisemmin alue sijaitsee, sitä monimutkaisemmaksi käy valintasignaalin muodostus. Puolittain kahden kahdeksan kilon lohkon alueella sijaitsevan kahdeksan kilon alueen (kuva 8.4c) valinta onnistuu vielä melko helposti "yksi kuudestatoista" -muuntimella ja TAI-piirillä.

Prosessori ja konekieli

Tarkastelkaamme nyt hieman mikroprosessorin toimintaa tietokoneessa. Käytämme esimerkkinä vanhaa 8-bittistä työhepoa 6502:ta, jollaisen voi löytää esimerkiksi Apple 2- ja VIC-20-koneista. Prosessoreiden perusrakenteet ovat hyvin samankaltaisia, joten tässä esitetyt periaatteelliset toiminnot löytyvät kyllä muistakin vastaavista piireistä.

Koska yksi kuva kertoo enemmän kuin yksi kilosana, niin siirrytäänpä heti aluksi analysoimaan yksinkertaistettua 6502:n kuvaa (kuva 8.5). Äsken esille



Kuva 8.5. Yksinkertaistettu 6502.

tulleet kolme väylätyyppiä, osoite-, data- ja ohjausväylä, ovat nyt näkyvillä kuvan laidoilla. Huomaa, että kuvan keskellä oleva ristin muotoinen sisäinen dataväylä ja varsinainen dataväylä ovat aivan eri asioita.

Lohko nimeltään rekisterit on varmasti sisällöltään tuttu kaikille konekieliohjelmoijille. Pienemmiksi paloiksi hajoitettuna sieltä nimittäin paljastuu koko joukko pieniä kahdeksan bitin muisteja, joissa lukuja pidetään väliaikaisesti tallessa. Näitä muisteja kutsutaan yleensä rekistereiksi ja eri rekistereille on annettu nimiksi esimerkiksi akku, X, Y ja ohjelmalaskuri.

Näistä viimeksi mainitulla on tarkoin määrätty tehtävä. Ohjelmalaskuri pitää jatkuvasti yllä 16-bittistä lukua, jonka osoittamassa muistipaikassa parhaillaan suoritettava käsky sijaitsee. Osoiterekisteri pitää yllä parhaillaan luettavan tai kirjoitettavan muistipaikan numeroa eli osoitetta. Pino-osoitin puolestaan sisältää erityisen pino-muistin rajan osoitteen.

Akku, X ja Y ovat yleiskäyttöisiä rekistereitä, joita konekielisen ohjelman käskyt käyttävät. Niihin voi esimerkiksi lukea muistissa olevia lukuja ja muunnella niitä sitten erilaisilla konekielen käskyillä. Tavallisia toimintoja ovat esimerkiksi yhteen- ja vähennyslasku, kahden luvun vertailu ja kahdella kertominen. X:n ja Y:n käyttö on huomattavasti akkua rajoitetumpaa, esimerkiksi yhteen- ja vähennyslasku ei ole mahdollista.

Laskutoimituksille on prosessorissa pyhitetty aivan oma lohkonsa. Sen nimi on 'aritmeettislooginen osa' eli englanninkielisistä sanoista 'arithmetic logic unit' lyhennettynä ALU. Sivistyssana aritmetiikka tarkoittaa suomeksi yksinkertaisesti laskuoppia. ALU suorittaakin kaikki prosessorin laskennalliset tehtävät. Lisäksi sen tehtäviin kuuluu ns. loogisten toimintojen suorittaminen. Näitä ovat esimerkiksi Basicistakin tutut TAI- ja JA- toiminnot (engl. OR ja AND).

Vaikka monet prosessorit pystyvät hämmästyttäviin laskentasuorituksiin, ei ALU:n toiminnassa ole mitään, mitä voisi kutsua älyksi. Siihen syötetään vain kaksi lukua ja tieto halutusta laskutoimituksesta. ALU:ssa luvut kulkevat sopivasti kytkettyjen loogisten porttien läpi, jonka jälkeen laskun tulos putkahtaa ulos.

ALU:ja valmistetaan myös erillisinä mikropiireinä, esimerkiksi LS-TTL -piiri nimeltään 74LS181 sisältää ALU:n, joka kykenee käsittelemään kahta neljäbittistä lukua 16 eri tavalla. Luvut voidaan esimerkiksi laskea yhteen, vähentää toisesta ykkönen jne. Piiri koostuu vain 67 perusportista. Useimpien prosessoreiden ALU:t ovat toki huomattavasti monipuolisempia kuin 74LS181, esim. 6502:n sisältämä käsittelee kahdeksanbittisiä lukuja ja monet muut selviävät nykyään 16 ja jopa 32 bitistä.

Tuon bittimäärän suuruus kertoo paljon prosessorin laskentanopeudesta. Kahdeksalla bitillä voidaan esittää 256 eri lukua, mikä useimpiin käytännön tehtäviin on aivan liian pieni määrä. Käytännössä suurempien lukujen lasku tapahtuukin hajoittamalla luvut pieniksi paloiksi ja laskemalla niillä vähän kerrallaan. Näinhän me ihmisetkin teemme laskiessamme isoilla luvuilla paperilla allekkain erikseen sadat, kymmenet jne.

Mitä pienemmiksi paloiksi luvut joudutaan pilkkomaan, sitä enemmän aikaa luonnollisesti kuluu koko laskuun. Niinpä onkin loogista, että ALU:n koon kasvu merkitsee aina nopeampaa tietokoneen toimintaa. Tässä on myös eräs selitys sille, miksi esimerkiksi 32-bittiset koneet ovat 8-bittisiä nopeampia.

Tarkastellaan seuraavaksi yhtä konekielistä käskyä ja sen aiheuttamaa liikennettä väylillä. Tehdään aluksi oletus että muistipaikat 49 152, 49 153 ja 49 154 sisältävät luvut 174, 32 ja 208, kuten listauksessa 1 on havainnollistettu. Oletetaan vielä, että osoitepuskurin 16 bittiä sisältävät aluksi noista osoitteista ensimmäisen (49 152) ja että ohjausväylän johtimet ovat sellaisessa tilassa, että ne määräävät muistia antaamaan em. osoitteen sisältävän luvun (174). Tällöin muisti uskollisena alamaisena lähettää tuon luvun dataväylää pitkin isännälleen.

Tullessaan prosessoriin luku siirtyy käskyrekisteriin ja edelleen käskynpurkajalle. Käskyrekisteri ei ylevästä nimestään huolimatta ole mitään muuta kuin

Osoite	Sisältö	Käsky
49151	...	
49152	174 032 208	LDX 53280
49155	...	

Listaus 1.

tavallinen kahdeksan bitin rekisteri, jossa parhaillaan suoritettava käsky pidetään siihen asti, kun seuraava haetaan muistista.

Käskynpurkaja puolestaan on laite, joka suorittaa lyhyitä yksinkertaisista toiminnoista koostuvia liikenteenohjaustehtäviä. Noiden tehtävien tarkempi sisältö riippuu parhaillaan suoritettavasta käskystä. Käytännössä käskynpurkaja toteutetaan usein muutamasta kiikusta tehdyllä laskurilla sekä pienellä ROM-muistilla, joka sisältää käskynpurkajan kullakin käskyllä prosessorin eri osille lähettämät signaalit. Noita signaaliryhmiä sanotaan myös prosessorin mikro-ohjelmakoodiksi.

Äsken osoitteesta 49 152 luettu luku 174 esimerkiksi määrää käskynpurkajalle seuraavanlaisen tehtävän:

1) Siirrä ohjelmalaskurin arvo (49 152) osoiterekisteriin yhdellä kasvatettuna. (Kasvatus voidaan prosessorista riippuen tehdä joko ALU:ssa tai suoraan rekisterin hyväksymällä kasvatussignaalilla.)

2) Aseta ohjausväylä siten, että muisti antaa osoiterekisterin osoittaman muistipaikan (49 153) arvon (32). Tallenna tuo arvo hetkeksi apurekisteriin.

3) Kasvata osoiterekisterin arvoa yhdellä.

4) Aseta ohjausväylä siten, että muisti antaa osoiterekisterin osoittaman muistipaikan (49 154) arvon (208).

5) Aseta äsken luetut kaksi lukua (8 + 8 = 16 bittiä) peräkkäin osoiterekisteriin.

6) Aseta ohjausväylä siten, että muisti antaa osoiterekisterin osoittaman muistipaikan (256 x 208 + 32 = 53 280) arvon.

7) Tallenna tuo arvo X-rekisteriin.

8) Lisää ohjelmalaskurin arvoon 3. (Seuraavan käskyn alkukohta).

Kuten 6502-konekieltä taitavat varmaan huomasivat, kyseessä oli käsky nro 174 eli LDX, joka lataa luvun muistista X-rekisteriin. Huomaamme, että aivan ruohonjuuritasolle purettuna tuo niin vähän tekevä käsky saakin aikaan melkoisen vipinän prosessorissa. Ohjelmoijan näkökulmasta katsoen prosessori ei tehnyt muuta kuin latasi X-rekisteriin muistipaikan 53 280 arvon.

Osoite	Sisältö	Käsky
49151	...	
49152	174 174 174	LDX 44718
49155	...	

Listaus 2.

Huomaa äskeisestä kuvauksesta myös viimeinen rivi, jolla ohjelmalaskuria kasvatetaan kolmella. Tuon toimenpiteen merkitys on siinä, että sen jälkeen ohjelmalaskuri sisältää seuraavan suoritettavan käskyn osoitteen, josta ohjelman suoritus jatkuu. Luku kolme tulee tarkastelemamme käskyn vaatimasta muistimäärästä. Lukuja luettiin kaikkiaan kolmesta eri kohdasta, joten seuraava käyttämätön on juuri kolmen päässä ensimmäisestä. Kaikkia prosessorin ymmärtämiä käskyjä varten on olemassa edellisen LDX:n kuvauksen kaltainen käskynpurkajan toimintosarja.

Kuten edellä huomasimme, koostui yksi käsky yhdestä tavusta eli kahdeksasta bitistä, joilla voidaan esittää 2 ^ 8 = 256 erilaista asiaa. 6502-prosessori ei kuitenkaan tunne kuin reilut 150 käskyä, joten jäljelle jää satakunta merkityksetöntä arvoa. Osa näistä saattaa käskyinä käytettynä sekoittaa prosessorin, osa tekee jotain lähes järjellistä. Jälkimmäisistä konekieli-ihmiset käyttävät nimitystä dokumentoimattomat käskyt. Ne ovat käskyjä, joista saattaa joskus olla hyötyä, mutta joita prosessorin valmistaja ei niiden epämääräisen tai epävarman toiminnan vuoksi ole tarkoittanut käytettäviksi.

Jos edellisessä esimerkissä muistissa olisikin listauksen 2 mukaisesti sekä 208:n että 32:n tilalla ollut 174, olisi toteutettava käsky edelleen ollut LDX, mutta luettavaksi osoitteeksi olisikin tullut (174 x 256 + 174 = 44 718). Ensimmäinen 174 merkitsee käskyä ja kaksi seuraavaa täsmentävät sitä määräämällä osoitteen, josta luku haetaan. Noita kahta kutsutaan myös käskynsä operandeiksi.

Muistissa ollessaan kaikki kolme lukua eivät mitenkään eroa toisistaan, mutta ohjelmana toimiessaan niillä on selvästi erilaiset merkitykset. Tämä voi aiheuttaa monia pulmatilanteita, jos prosessori jossain vaiheessa lukee ohjelmansa väärästä kohtaa muistia. Jos esimerkiksi listaus 2 jatkuu listauksen 3 mukaisesti kahdella nollalla (nolla on BRK-käsky, joka keskeyttää ohjelman), tapahtuisi ohjelman ajossa seuraavaa:

1) Lue X-rekisteriin arvo muistipaikasta 44 718.

2) Keskeytä ohjelma.

Osoite	Sisältö	Käsky
49152	174 174 174	LDX 44718
49155	000	BRK
49156	000	BRK

Listaus 3.

Osoite	Sisältö	Käsky
49152	174	...
49153	174 174 000	LDX 174
49156	000	BRK

Listaus 4.

Listauksissa 3 ja 4 on muistin sisältö täsmälleen sama (174, 174, 174, 0, 0), mutta jälkimmäisessä ohjelman tarkastelu on aloitettu yhtä muistipaikkaa myöhemmin. Ensimmäinen merkitsevä luku (käsky) on tällöin edelleen 174, mutta sen operandeiksi tulevatkin luvut 174 ja 0, eli osoitteeksi kaikkiaan 0 x 256 + 174 = 174. Sanallinen kuvaus ohjelmasta 4 olisi siis:

1) Lue X-rekisteriin arvo muistipaikasta 174.

2) Keskeytä ohjelma.

Kuten huomaamme, tekevät nuo kaksi ohjelmaa aivan eri asioita. Ohjelman oikean aloituskohdan valitseminen onkin prosessorin äidinkielellä eli konekielellä tehdyillä ohjelmilla vielä tärkeämpää kuin esimerkiksi Basicissa, jossa vaikkapa RUN 100 -käskyllä aloitettu ohjelma jättää vain ohjelman ensimmäiset käskyt (ennen riviä 100) suorittamatta, mutta jatkaa kyllä sitten lopun ohjelmaa oikein.

Kaikista toiminnoistaan ja ominaisuuksistaan huolimatta prosessori sisältää loppujen lopuksi vain läjän kiikuista kasattuja rekistereitä, kombinaatiopiirejä (ALU), sekä näille ulkoa saatujen käskyjen tahdissa ohjaussignaalisarjoja antavan käskynpurkajan, joka sekin on vain kasa kiikkuja ja muistia.

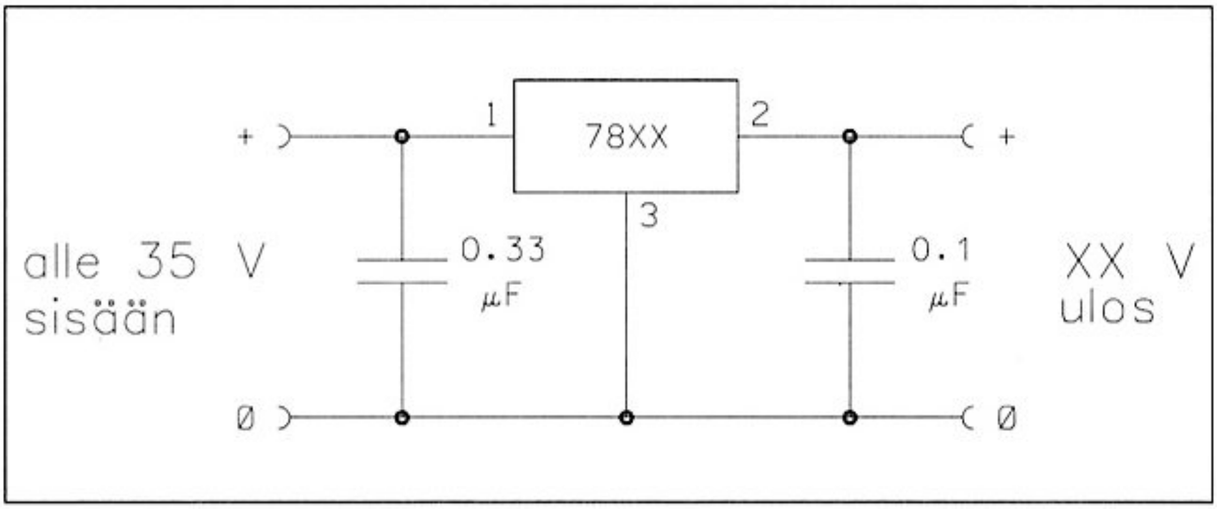
Virtalähteet ja jännitteen vakavointi

Kaikki elektroniset laitteet vaativat toimiakseen sähköä. Harrastajarakentelija tarvitsee laitteilleen usein kohtuullisen pieniä noin 5–24 voltin jännitteitä. Tällaisia jännitteitä tarjoavia virtalähteitä on saatavilla valmiina kaupoista lukuisia erilaisia.

Digitaalielektroniikassa käytetään hyvin usein 5 voltin käyttöjännitettä ja laitteiden toiminnan kannalta on erittäin tärkeää, että jännite pysyy mahdollisimman lähellä tätä. Sallittu poikkeama on esimerkiksi TTL-piiriperheellä vain 0,25 volttia suuntaan tai toiseen. Liian pieni käyttöjännite tekee laitteen toiminnasta epävarman ja liian suuri rikkoo herkät mikropiirit.

Näin tarkka jännite voidaan luotettavasti saada vain ns. vakavoidusta teholähteestä. Tavallisten halpojen virtalähteiden, sellaisten joita käytetään esimerkiksi matkaradioiden paristojen korvikkeena, jännite yleensä vaihtelee jopa useita voltteja verkkojännitteen vaihteluiden ja virtalähteestä otetun virran mukaan.

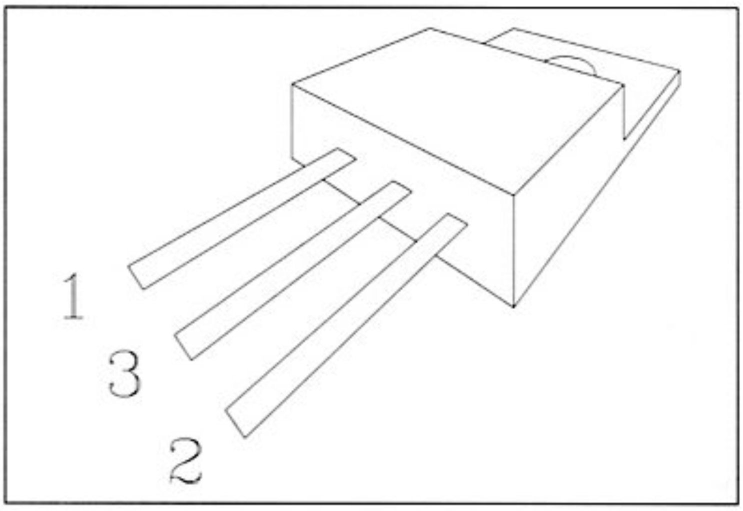
Jännitteen vakavointi onnistuu helpoiten tätä varten valmistettavilla edullisilla mikropiireillä. Erittäin yleinen on ns. 7800-mikropiirisarja. Tämän sarjan piirejä käyttämällä tarvitaan kuvan 1 mukaisesti vain pari lisäkondensaattoria.



Kuva 1. 7800-sarjan jännitevakaajan peruskäyttökytkentä.

Piiri	Antojännite
7805	5 V
7806	6 V
7808	8 V
7812	12 V
7815	15 V
7818	18 V
7820	20 V
7824	24 V

Taulukko 1. 7800-sarjan regulaattorit.



Kuva 2. TO-220-kotelon johdinjärjestys.

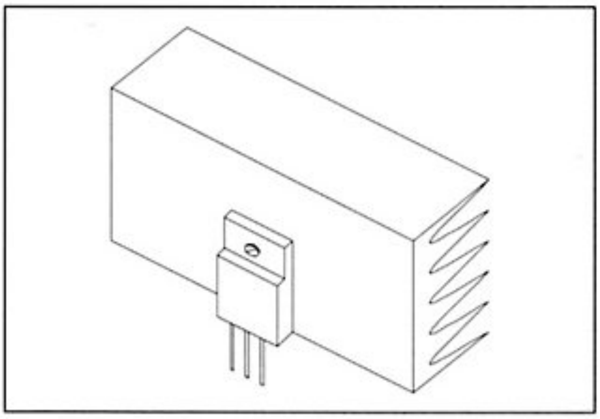
Sarja	Antovirta
78xx	1 A
78Mxx	0.5 A
78Sxx	2 A

Taulukko 2. 7800-rinnakkaissarjat.

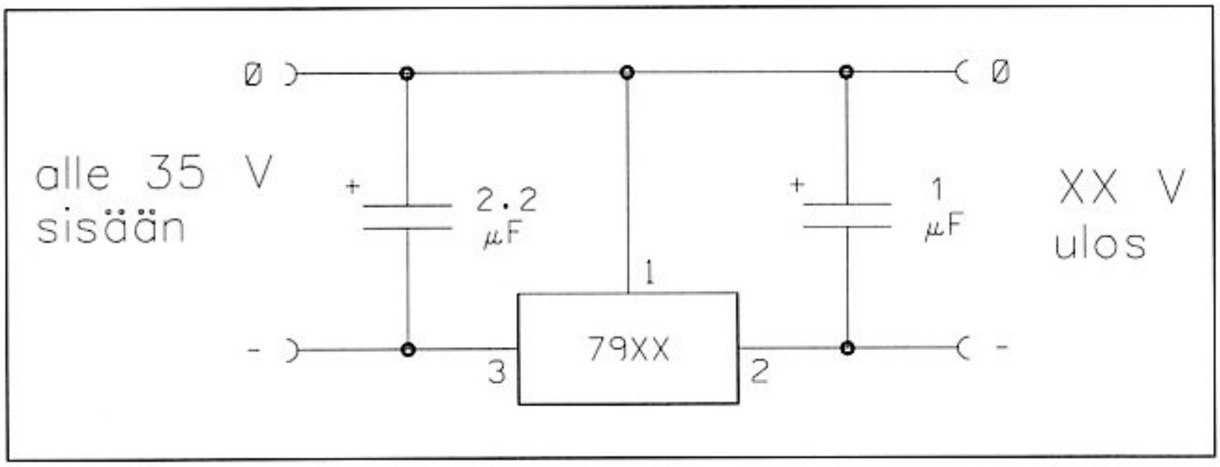
Kuvan 1 vakavointipiiriltä saadaan ulos vakiosuuruinen jännite. Jännitteen suuruus riippuu käytetystä mikropiiristä. Oikean piirin valinta on helppoa sillä sarjan piirien vakavointijännitettä käytetään suoraan nimen kahtena jälkimmäisenä numerona. Siten esimerkiksi 5 voltin vakavointipiirin, tai regulaattorin, nimi on 7805. Koko sarja on esitelty taulukossa 1.

7800-sarjan piiristä saa ottaa enintään 1 ampeerin virran. Piirit on sisäisesti suojattu annon oikosulkua ja ylikuumenemista vastaan. Sarjan piireistä on saatavilla myös 0,5 ja 2 ampeerin versiot. Poikkeava maksimivirta ilmaistaan 78- ja jännitenumeroiden välissä olevalla kirjaimella M tai S taulukon 2 mukaisesti. Kahden ampeerin 12 voltin piirin olisi siis 78S12.

7800-regulaattorin TO-220-kotelon ulkomuoto ja jalkajärjestys on esitetty kuvassa 2. Huomaa, että jalkojen numerointi ei etene aivan loogisesti. Regulaattorit tuottavat toimiessaan runsaasti lämpöä, joka pitää saada johdettua pois piiristä. Tähän käytetään yleensä alumiinista valmistettua jäähdytysripaa, johon piiri kiinnitetään ruuvaamalla. Kuva 3 esittää TO-220-kote-



Kuva 3. TO-220-kotelon jäähdytys.

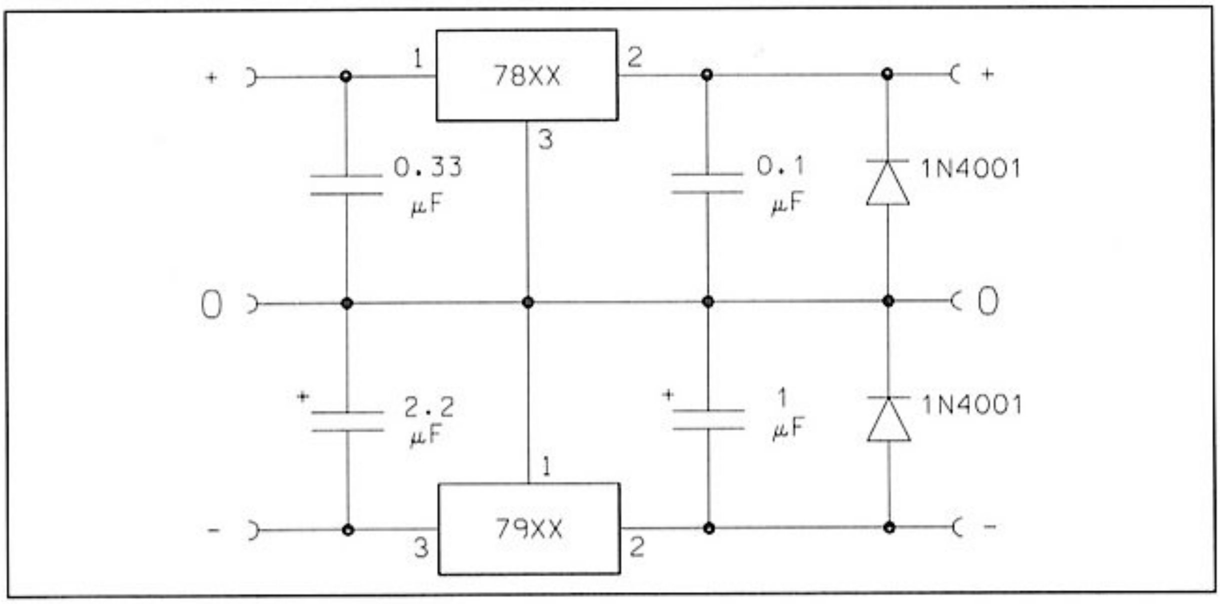


Kuva 4. 7900-sarjan jännitevakaajan peruskäyttökytkentä.

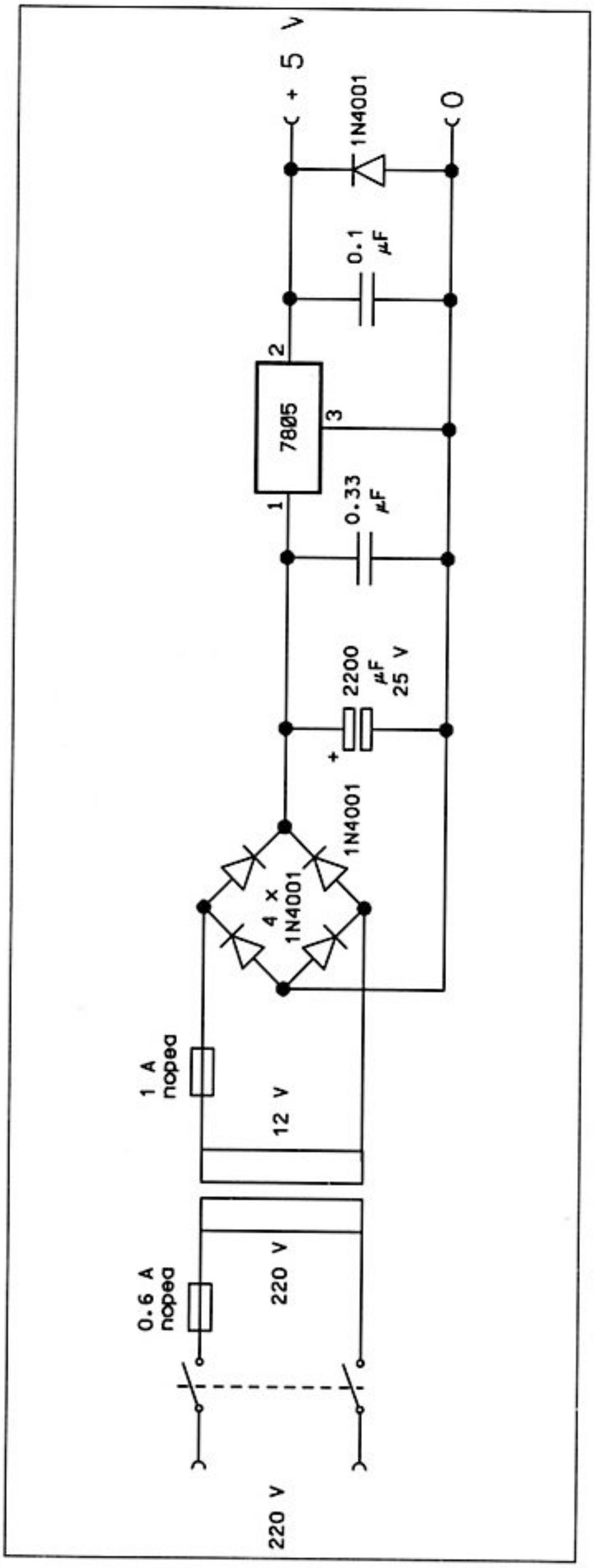
loisen piirin kiinnitystä tyypilliseen jäähdytysripaan.

Jos lämmön siirtymistä piiristä jäähdytysripaan halutaan tehostaa, voidaan piirin ja rivan väliin levittää piirasvaa. Monesti jäähdytysrivan ja piirin väliin vielä asetetaan hyvin lämpöä johtava kiillelevy, joka eristää rivan ja piirin sähköisesti toisistaan. Tämä on usein välttämätöntä, koska piirikoteloiden metalliset kiinnityskorvakeet on yleensä kytketty johonkin piirin kiinnitysnastoista, esimerkiksi 7800-sarjan piireissä korvake on kytketty nastaan 3. Jos samaan jäähdytysripaan kiinnitetään useita erilaisia regulaattoreita ilman kiillelevyjä, voisi rivan kautta syntyä ei-toivottu kytkentä piirien kesken.

Negatiivista jännitettä voidaan vakavoida 7900-piirisarjalla. Tämä on aivan 7800:n kaltainen sarja. Peruskytkentä on esitetty kuvassa 4. Ainut ero 7800:n kytkentään on kondensaattoreiden arvojen muuttuminen ja piirin jalkojen järjestyksen vaihtuminen.



Kuva 5. Vakavoitu kaksoisjännitelähde.



Kuva 6. 5 V/1 A vakavoitu verkkolaite.

Toisinaan laitteisiin vaaditaan ns. symmetrinen virtalähde, joka antaa sekä positiivisen että negatiivisen jännitteen. Tällainen virtalähde on helppo toteuttaa 78- ja 79-piiriparilla kuvan 5 mukaisesti.

7800- ja 7900-sarjan piirit vaativat toimiakseen nimellisjännitettään vähintään pari volttia suuremman syöttöjännitteen. Jotta siis esimerkiksi 7805:n regulaattorista saataisiin todella 5 voltin jännite ulos, pitää siihen syöttää vähintään noin 7 volttia sisään. Tarpeettoman suurta jännite-eroa on syytä välttää, sillä se vain aiheuttaa piirin lämpiämistä.

Regulaattoripiirille syötettävä jännitekin pitää tietenkin saada jostain. Turvallisuussyistä varsinkin aloittelevan rakentelijan kannattaa ehdottomasti käyttää tähän tarkoitukseen vain valmiita kaupasta ostettuja hyväksyttyjä virtalähteitä. Virtalähdettä hankkiessaan kannattaa varmistua, että se jaksaa antaa vaadittavan jännitteen lisäksi myös riittävästi virtaa. Hieman enemmän virtalähteeseen sijoittamalla voi ostaa valmiina esimerkiksi kuvassa 6 esitetyn kaltaisia vakavoituja virtalähteitä.

A/D-muuntimen sovelluksia

Digitaalinen signaalinkäsittely valtaa jatkuvasti alaa perinteisesti analogiatekniikalla toteutetuissa laitteissa. Tällaisia uuden tekniikan laitteita ovat esimerkiksi digitaaliset vahvistimet, CD-soitin, digitaaliset kuva- ja ääninauhurit sekä erilaiset mikroihin liitettävät mittalaitteet.

Tässä esitettävän lisälaitteen avulla saadaan mikrotietokone muunnettua erilaisten analogiasignaalien, kuten puheen ja musiikin käsittelijäksi ja oskilloskoopiksi.

Oheisten sovellusohjelmien avulla voidaan koneen muistiin tallentaa puhetta tai musiikkia ja toistaa se laitteen kautta suoraan kuulokkeille tai vahvistimelle. Toiston nopeutta voi muuttaa ja toiston voi suorittaa vaikkapa takaperin. Ohjelmia voi itse kukin halutessaan täydentää erilaisilla lisätoiminnoilla. Esimerkiksi kaikuefekti saadaan aikaan summaamalla ajallisesti vanhempia näytteitä toistettavaan signaaliin.

Analogiasignaalilla tarkoitetaan signaalia, joka voi saada portaattomasti minkä tahansa arvon minimi- ja maksimiarvojensa välillä. Tällainen signaali on esimerkiksi vahvistimelta kaiuttimelle menevä vaihtojännite.

Signaalin digitoinnilla tarkoitetaan signaalin hetkellisarvon ilmaisemista numeroarvona. Muunnoksen suorittavan analogia-digitaalimuuntimen ominaisuudet määräävät sen, miten pieni signaalitason muutos on havaittavissa. Digitoitu arvo muuttuu aina portaissa vaikka analogiasignaali muuttuisikin tasaisesti.

Digitointiportaan suuruus riippuu siitä, miten monella numerolla digitoitu luku ilmaistaan. Esimerkiksi 8-bittisellä muuntimella voidaan digitoitavasta signaalista saada 256 erilaista arvoa. Tämä määrä erilaisia arvoja riittää useimpiin sovelluksiin. Esimerkiksi nykyisin myytävissä muistioskilloskoopeissa käytetään yleisesti 8-bittisiä muuntimia.

On tärkeää, että A/D-muunnoksessa muuntimen toiminta-alueesta käytetään mahdollisimman suuri osa hyväksi. Tämä tapahtuu käyttämällä muuntimen edessä aseteltavaa vaimenninta tai vahvistinta, jolla muunnettavan signaalin vaihteluväli sovitetaan mahdollisimman tarkasti muuntimen ymmärtämän alueen sisään.

Näytteenottotaajuus

Mikäli vaihtojännitteen aaltomuodosta halutaan kuva digitoitujen näytteiden perusteella pitää jännitteen hetkellisarvon mittaus toistaa useita kertoja yhden signaalijakson aikana. Jotta signaalin alkuperäinen aaltomuoto olisi palautettavissa digitoitujen näytteiden perusteella tulee näytteenottotaajuuden olla vähintään kaksi kertaa niin suuri kuin suurin muunnettava taajuus. Muunninpiirillä, jonka näytteenottotaajuus on 10 kHz voidaan näin käsitellä signaalia, joka sisältää taajuuksia 0...5 kHz. Tämä riittää hyvin puheelle ja musiikillekin mikäli ei tavoitella aivan hifi-tasoista toistoa.

Mikäli muunnettavan signaalin taajuus on näytteenottotaajuuden puolikasta suurempi tapahtuu taajuuksien sekoittumista, ns. alias-ilmiö. Tällöin signaalia toistettaessa ei saadakaan tuloksena alkuperäistä taajuutta vaan näytteenottotaajuuden ja muunnetun taajuuden erotus. Esimerkiksi 6 kHz taajuus muuttuu 10 kHz näytteenottotaajuudella 4 kHz:ksi. Tämän ilmiön estämiseksi käytetään äänentoistolaitteissa A/D-muuntimen edessä jyrkkäreunaista alipäästösuodatinta (antialiasing-suodatin), jolla estetään näytteenottotaajuuden puolikasta suurempien taajuuksien pääsy muuntimelle.

Puheessa ja musiikissa yli 5 kHz taajuiset signaalikomponentit ovat amplitudiltaan kuitenkin melko pieniä ja niistä johtuvat häiriöt vähäisiä. Tämän vuoksi laitteen yksinkertaistamiseksi on alipäästösuodatin muuntimen edestä jätetty pois.

Laitteen liitäntä mikroon

Itse laite sisältää säädettävällä vahvistimella varustetun A/D-muuntimen, joka liitetään mikroon. Tässä esitetään laitteen liittäminen PC-yhteensopivan mikron kirjoitinporttiin sekä Commodore 64:n käyttäjänporttiin. C64:n käyttäjäporttiliitosta soveltamalla laite on melko helppo siirtää useimpiin muihin kotimikroihin, esimerkiksi peliohjainliittimiin.

Laitteeseen kuuluu lisäksi päinvastaisen toiminnon suorittava D/A-muunnin, jonka avulla on mahdollista kuunnella digitoitua signaalia digitoinnin aikana sekä jälkeenpäin koneen muistista.

A/D-muuntimen avulla muunnetaan tutkittava analogiasignaali 8-bittisiksi luvuiksi, jotka tallennetaan koneen muistiin. Kun käytettävissä oleva muistialue on täyttynyt pysäytetään A/D-muunnin ja muistiin tallennettuja lukuja aletaan syöttää D/A-muuntimelle.

A/D-muuntimen rakenne

Muunnin on pyritty rakentamaan siten, että se soveltuu liitettäväksi mahdollisimman moniin mikroihin.

Laiteen sisäänmeno on varustettu 4-asentoisella herkkyyden säätimellä. Säätimen ollessa asennossa 1 on muunnin herkimmillään ja soveltuu esimerkiksi mikrofonista saatavan signaalin käsittelyyn. Tässä asennossa ottojännitteen 20 mV:n muutos saa muuntimen annon muuttumaan $00:sta $FF:ään. Muiden asentojen vastaavat jännitearvot ovat 200 mV, 2 V ja 20 V.

Laitteessa on erillinen ottoliitin vaihtojännitteelle (AC) ja tasajännitteelle (DC). Vaihtojänniteottoa käytetään äänitaajuisille signaaleille sekä muulloinkin kun halutaan suodattaa ottosignaalista pois siinä oleva tasajännite.

Tasajännitekytkettyä ottoa käytetään, kun halutaan digitoida tasajännitteitä tai erittäin pienitaajuisia vaihtojännitteitä.

Operaatiovahvistinta A1 käytetään A/D-muuntimen ottovahvistimena. Asteen vahvistukseksi on asetettu 100.

Analogiakytkin 4066 yhdessä kondensaattorin C3 ja operaatiovahvistimen A2 kanssa muodostaa näytteenotto- ja pitopiirin, mikä pitää A/D-muuntimen ottojännitteen vakiona muunnostapahtuman aikana.

Muunnoksen käynnistyspulssin aikana menee analogiakytkimen ohjausnasta 5 hetkeksi ylös. Tällöin ottovahvistimen A1 lähdössä oleva jännitetila siirtyy kytkimen kautta kondensaattoriin C3, jossa varaus säilyy A2:n suuren ottoimpedanssin ansiosta muuttumattomana koko muunnostapahtuman ajan.

D/A-muunnin

Digitaali-analogiamuunnin sisältää puskuripiirit IC4 ja IC5, joiden tehtävänä on muuntaa tietokoneen antoportista saatavat TTL-tasoiset jännitteet +5 V:n ja 0 V:n jännitetasoiksi. D/A-muunnin koostuu vastuksista R18...R25 ja operaatiovahvistimesta A3. Vastuksista on muodostettu binäärisesti painotettu summauskytkentä, jossa eniten merkitsevää bittiä vastaa pienin vastus.

Summauspisteenä toimii operaatiovahvistimen A3 invertoiva sisäänmeno. Vahvistimen lähdöstä saadaan digitaalisanaan verrannollinen jännite. Vahvistimen ei-invertoiva otto on asetettu käyttöjännitteen puoliväliin vastusjännitteenjakajalla R26, R27. Piirin kokonaisvahvistus määräytyy takaisinkytkentävastuksen R28 suhteena vastusten R18...R25 rinnankytkennän arvoon. Täten laskettu

vahvistuksen arvo on n. 0,2. Digitaalisanan muutos $00:sta $FF:ään aiheuttaa näin ollen antojännitteen muutoksen 0,2 x 5 V = 1 V.

Antojännite viedään alipäästösuodattimen A4 kautta kuulokeliittimelle. Alipäästösuodatin vaimentaa näytteenotosta johtuvaa 10 kHz signaalia. Piirin antoon voidaan liittää yli 30-ohmiset kuulokkeet.

Muunninpiirin toiminta

Laitteessa on käytetty 8-bittistä, helposti saatavaa muunninpiiriä ADC0804, jonka hinta on ostopaikasta riippuen 25–45 mk.

Piirin analogiapuolella on kaksi ottonastaa V+in ja V–in sekä referenssijännitenasta Vref. Piiri muuntaa V+in ja V–in-nastojen välisen jännite-eron digitaalisanaksi muuntimen antoon. Muuntimen lähdöstä saadaan digitaalisana $00 kun muuntimen ottonastojen välinen jännite-ero on 0 V. Piiri antaa lähtöihinsä sitä suuremman binääriluvun mitä suurempi positiivinen jännite V+in -nastassa on V–in-nastaan verrattuna. Kun jännite-ero on 2xVref on muuntimen lähdössä luku $FF.

Jos piirin Vref-nasta jätetään avoimeksi asettuu se piirin sisäisen jännitteenjakajan ansiosta käyttöjännitteen puoliväliin (2,5 V). Näin suuri referenssijännite edellyttää, että muunnettavan jännitteen tulisi vaihdella alueella 0–5 V, jotta piirin koko toiminta-alue $00–$FF tulisi käytettyä hyväksi. Näin suuren ohjausjännitteen saaminen on kuitenkin mahdotonta käytettäessä operaatiovahvistinta TL084 piirin ottovahvistimena.

Vahvistimen rakenteesta johtuen sen antojännite ei suostu menemään 1,5 V:n alapuolelle eikä juurikaan 4 V:n yläpuolelle. Tämän vuoksi kytkennässä pudotetaan ulkoisella 1k trimmerillä R6 referenssijännite 1 V:iin. Tällöin täysi lukema $FF saadaan kun piirin ottojen V+in ja V–in välinen jännite-ero on 2 V.

A/D-muuntimen V–in-nastan jännitetaso on nostettu vastusjännitteenjakajan (R8, R9, R10) avulla 2 V:iin. Potentiometrillä P1 säädetään V+in-nastan tasajännitetasoa.

Muunninpiirin kontrollisignaalit

Muunnostapahtuma käynnistyy kun WR-nasta käy alhaalla. Muunnoksen aikana nastan tulee olla ylhäällä.

Kun piiri on suorittanut muunnoksen menee INTR-nasta alas. INTR-nasta palaa ylös kun uusi muunnos käynnistetään vetämällä WR-nasta hetkeksi alas.

Piirin data-annot saadaan suuri-impedanssiseen tilaan nostamalla RD-nasta ylös. Tämä on tarpeen mikäli mikron samaa dataporttia käytetään ottona ja antona.

Invertteri (IC5) on kytketty piiriin siten, että piirin lähdöt menevät suuri-impedanssiseen tilaan, kun muunnoksen käynnistysnasta WR jätetään alas.

Laitteessa on WR-nastaan kytketty painike S2, jolla muunnos voidaan käynnistää manuaalisesti. Tämä on hyödyllinen nappi muuntimen toimintaa ja ohjelmaa testattaessa. Muunnin saadaan toimimaan itsenäisesti ilman mikroa yhdistämällä WR- ja INTR-nastat toisiinsa.

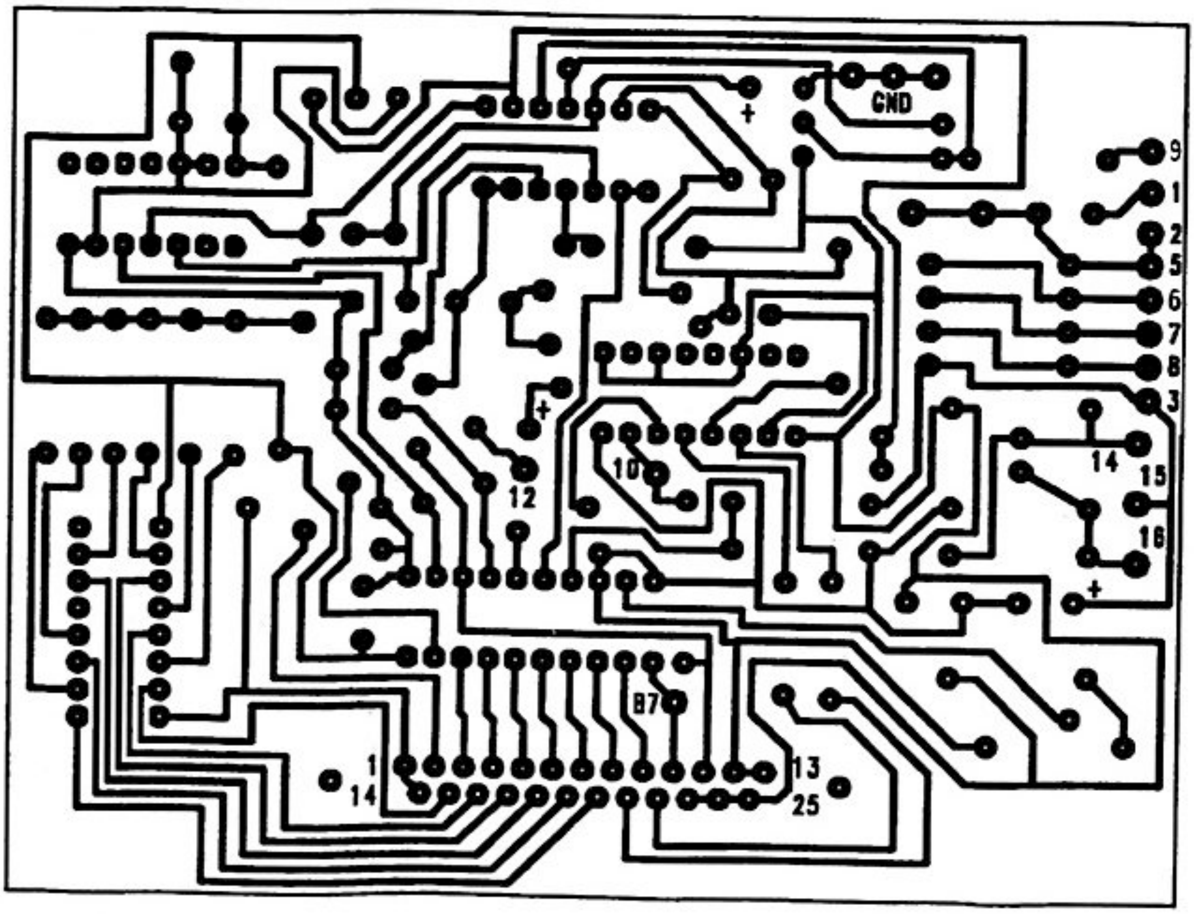
Piiri sisältää kello-oskillaattorin, jonka toimintataajuuden määrää nastojen 19 ja 4 välille kytketty RC-piiri. Muunnos vaatii noin 70 kellojakson ajan. Täten 700 kHz:n kellotaajuudella saavutetaan noin 10 kHz:n näytteenottotaajuus.

Kellotaajuutta voidaan nostaa pienentämällä kondensaattoria C4 tai vastusta R13. Jos RC-aikavakiota pienennetään liikaa lakkaa kello-oskillaattori värähtelemästä ja piiri lopettaa toimintansa.

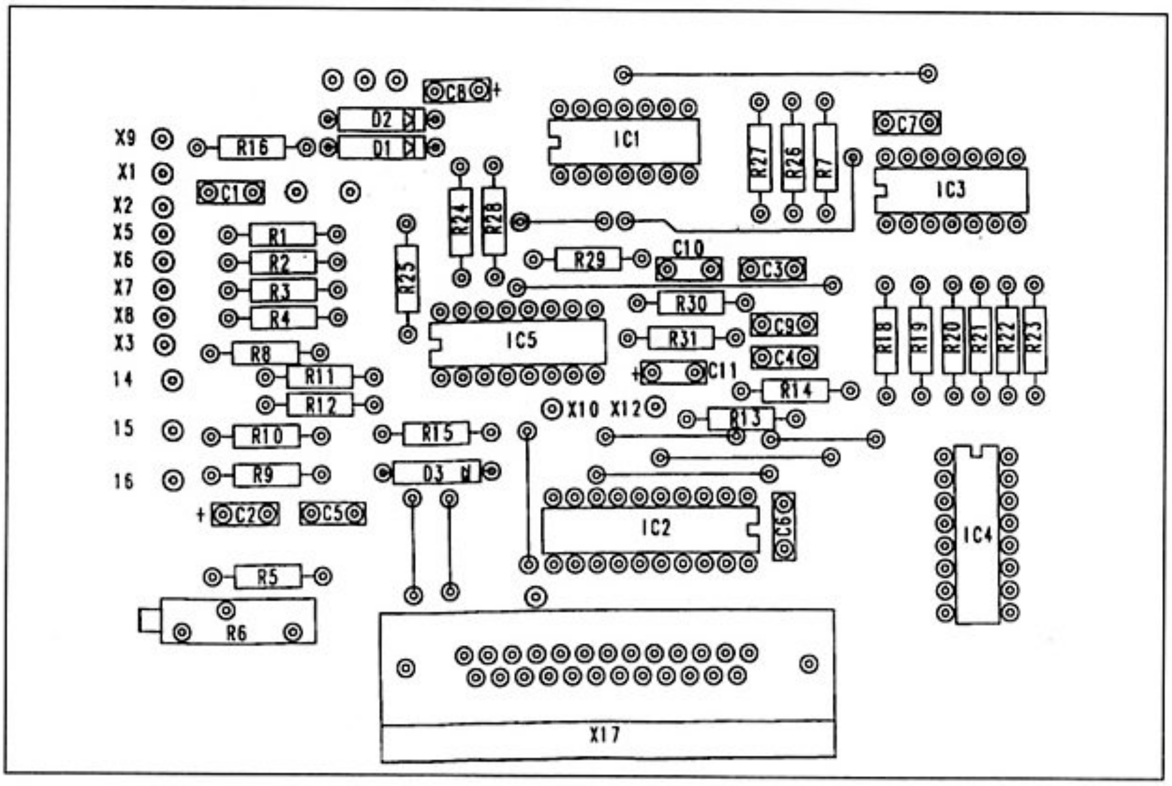
Muunnin toimii ns. peräkkäisapproksimaatioperiaatteella. Piiri sisältää D/A-muuntimen, jonka antojännitettä verrataan muuntimen ottojännitteeseen. Ensimmäisen vertailun aikana muunnin tutkii, onko ottojännite muunnettavan jännitealueen puolivälin ylä- vai alapuolella. Mikäli jännite on vertailutason yläpuolella tulee ylimmäksi bitiksi 1, muussa tapauksessa 0. Ensimmäisen vertailun tuloksen perusteella seuraavan vertailun aikana ottojännitettä verrataan kyseisen ylemmän tai alemman puolikasalueen puoliväliin. Tuloksen perusteella määräytyy toiseksi eniten merkitsevä bitti. Näin menetellen aina jäljelläolevan alueen puoliväliin vertaamalla tiedetään kahdeksannen vertailun jälkeen yhden bitin tarkkuudella ottojännitteen taso.

Rakentaminen

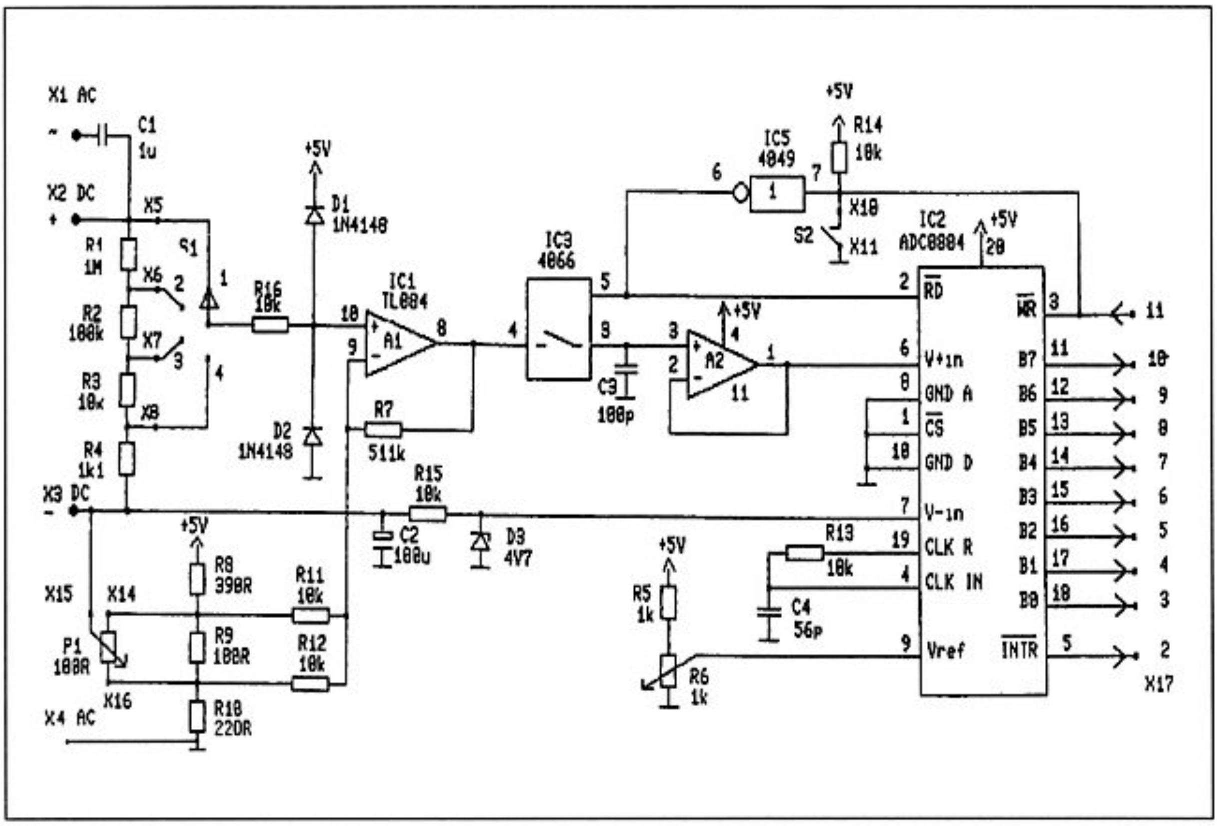
Kuvissa 1 ja 2 on esitetty laitteen piirilevy juotospuolelta nähtynä sekä komponenttien sijoittelu päältäpäin nähtynä. Kytkentäkaavio on esitetty kahteen osaan jaettuna kuvissa 3 ja 4. On hyvä huomioida, että A/D-muunninpiiri IC2 on eri päin kuin muut levyllä olevat piirit (lovi oikealle, muissa vasemmalle). Väärinpäin asennettu piiri tuhoutuu suurella todennäköisyydellä. Tantaalikondensaattorien ja diodien napaisuuden kanssa kannattaa myös olla tarkkana.



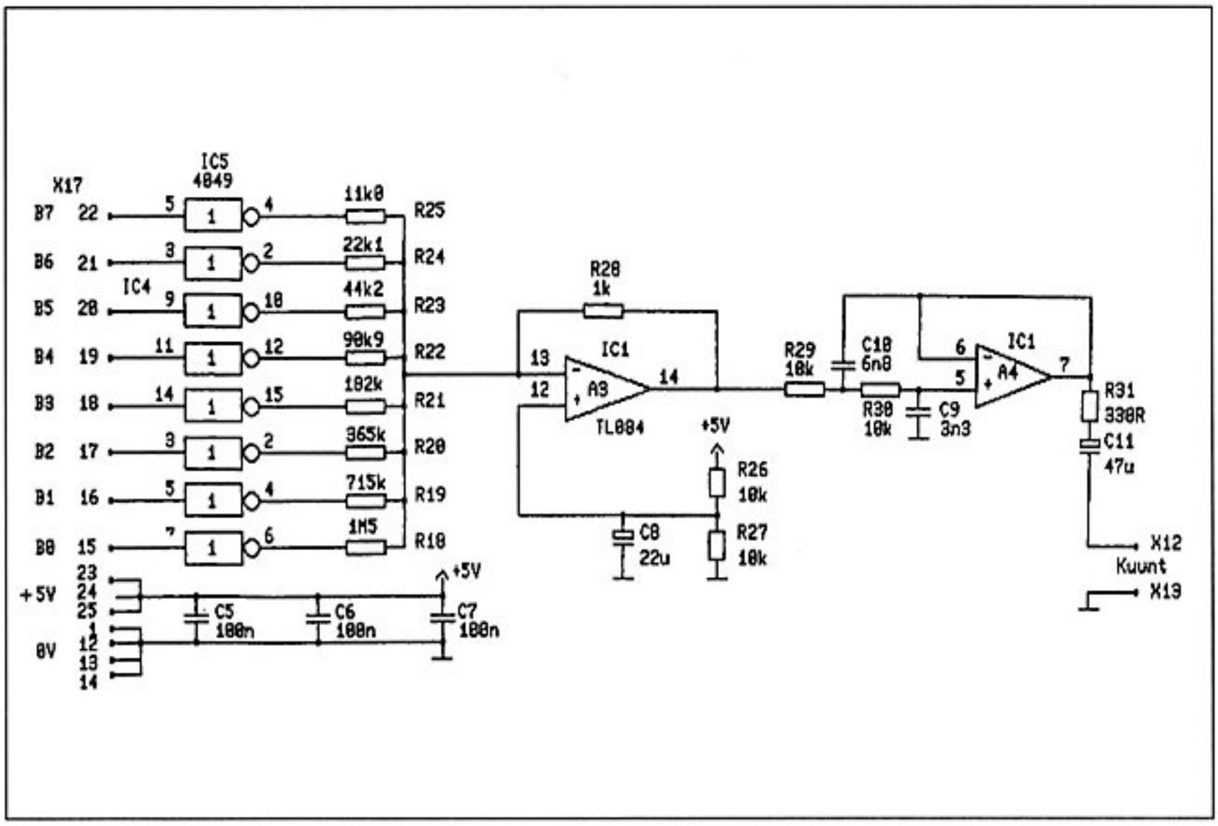
Kuva 1. Muuntimen piirilevy juotospuolelta.



Kuva 2. Muuntimen piirilevyn osasijoittelu.



Kuva 3. Muuntimen A/D-osan kytkentäkaavio.



Kuva 4. Muuntimen D/A-osan kytkentäkaavio.

Tantaalikondensaattorien jännitekeston tulee olla vähintään 5 V. Suurempi jännitekesto ei ole haitallista. Tantaalikondensaattorit voi korvata vastaavan kapasitanssiarvon omaavilla elektrolyyttikondensaattoreilla.

Muuntimen otot on suojattu vahingossa ottoliittimiin kytkettävää ylijännitettä vastaan osilla R15, R16, D1, D2 ja D3, joiden tehtävänä on rajoittaa mikropiireille tulevat jännitteet suunnilleen laitteen käyttöjännitteen suuruisiksi.

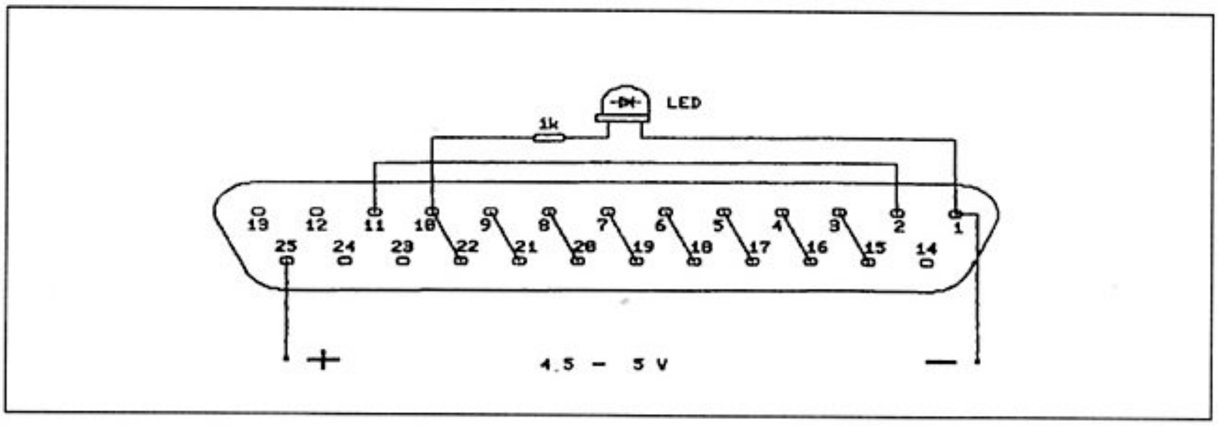
Toiminnan testaus

Testausta varten tehdään 25-napaiseen D-liittimen pistokkeeseen kuvassa 6 esitetyt kytkennät. Testausta varten kytketään laitteelle käyttöjännite esimerkiksi 4,5 V:n taskulampun paristosta. Trimmerillä R6 säädetään referenssijännitteen tasoksi 1,0 V. Jos käytettävissä ei ole yleismittaria, säädetään trimmeri R6 suunnilleen puoliväliin.

S2-nappia painamalla muuntimen pitäisi käynnistyä ja jatkaa toimintaansa itsenäisesti, jolloin voidaan tutkia piirin antojen tiloja joko yleismittarilla tai ledin avulla, jonka kanssa sarjaan on kytketty noin 1 k:n vastus. Tällainen sarjakytkentä on piirretty testausliittimen kytkentäkuvaan (kuva 6) eniten merkitsevän bitin antonastan 10 ja maan 1 välille. Ledin katodijohdon puolella on ledin reunuksessa pieni viiste. Tämä johto kytketään laitteen maahan.

Tasonsäätöpotentiometriä P1 kierrettäessä pitäisi jossain vaiheessa eniten merkitsevään data-antoon kytketyn ledin syttyä ja takaisinpäin kierrettäessä jälleen sammua. Muuntimen paras toimintapiste on kohdassa, jossa ledi on juuri syttymäisillään. Tällöin muuntimen lähdössä on luku $80 ja muuntimella on maksimi toimintavara kumpaankin suuntaan.

Kun nyt muuntimen liittimien X12 ja X13 välille kytketään kuuloke pitäisi siitä kuulua muuntimen aiheuttamaa pientä pohjakohinaa ja kun herkkyyskyt-



Kuva 6. Muuntimen testausliitin.

kin käännetään asentoon 1 ja sormella kosketetaan ottoliitintä 1 pitäisi kuulokkeesta kuulua hurinaa.

Kun AC-ottoliittimien X1 ja X4 välille tuodaan signaali esimerkiksi radion tai nauhurin kuulokeliittimestä, niin muuntimen digitoima ja jälleen analogiseen muotoon muuntama signaali pitäisi kuulua kuulokkeista. Herkkyydenasetuskytkintä S1 kiertämällä valitaan sopiva voimakkuustaso siten, että muunnin ei yliohjaudu ja ääni kuuluu säröytymättä.

Tämän jälkeen muunnin liitetään mikroon ja toiminnan testaus ohjelman kanssa voi alkaa.

Muuntimen DC-ottosignaalin miinusliitin X3 ei ole muuntimen eikä mikron maapotentiaalissa vaan noin 2 V:n jännitteessä maahan nähden. Jos signaalilähteenä toimivan laitteen maa on muuta kautta yhdistetty mikron ja täten myös muuntimen maahan, liitin X3 jätetään avoimeksi. Tällöin DC-ottosignaali kytkeytyy liittimen X2 ja laitteen maan (X4) välille.

Tärkeää on huomata, että A/D-muunnin voi toimia vain kun V+in on positiivisempi kuin V–in. V+in jännitteen tulee muuttua alueella 2–4 V, kun potentiometriä P1 kierretään.

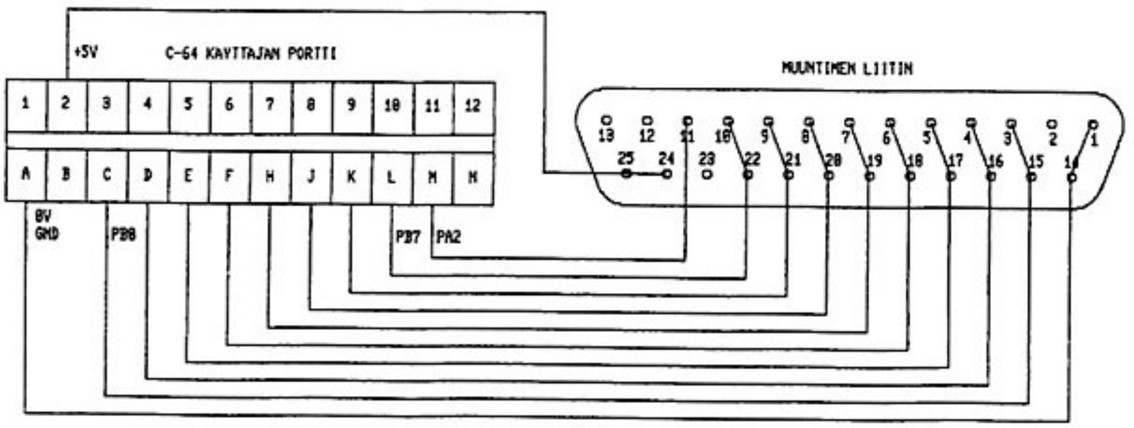
Jos laite ei toimi halutulla tavalla voi syynä olla katkokset tai tinasillat piirilevyllä, puuttuva hyppyjohdin tai puuttuva, väärä tai viallinen komponentti, väärinpäin kytketty diodi, tantaali tai mikropiiri. Tärkeää laitteen hyvän toiminnan kannalta on se, että kaikki 8 bittiä varmasti menevät mikrolle asti ja tulevat sieltä takaisin D/A-muuntimille oikeisiin nastoihin.

Käyttö C64:n kanssa

C64:n äänendigitointiohjelma on esitetty konekielisenä listauksessa 1. Saman ohjelman sisältävä Basic-ohjelma on listauksessa 2.

A/D-muunnostulosten tallennuksen alkuosoite sijoitetaan C64:ssä 0-sivun osoitteisiin $FB ja $FC. Muunninta ohjataan käyttäjänportin kautta kuvan 5 mukaisen kaapelin kautta. PA2-signaalilla käynnistetään A/D-muunnos. PA2 asetetaan annoksi asettamalla osoitteen $DD02 bitti 2 ykköseksi. Data saadaan sisään B-portista, joka asetetaan otoksi kirjoittamalla osoitteeseen $DD03 luku $00.

Muunnin käynnistyy ohjelman kohdassa R_SEUR. Tällöin osoitteen $DD00 bitti 2, joka vastaa PA2-antoa, käy alhaalla. Muuntimelta odotetaan tulosta viivealiohjelmassa R_VIIVE, jossa x-rekisteriin asetetaan viiveen määräävä luku. Viive on kokemusperäisesti haettu sellaiseksi, että muunnin ehtii suorittaa muunnoksen loppuun viiveen aikana.



Kuva 5. Muuntimen ja C64:n välinen kaapeli.

Mikäli muuntimen kellotaajuus on normaalia pienempi pitää viivettä kasvattaa. Muunnoksen tulos luetaan B-portin osoitteesta $DD01. Tulos tallennetaan osoitteeseen, joka saadaan osoitteissa $FB ja $FC olevan luvun ja Y-rekisterin summana. Y:n arvoa kasvatetaan yhdellä jokaisen muistiintalletuksen jälkeen.

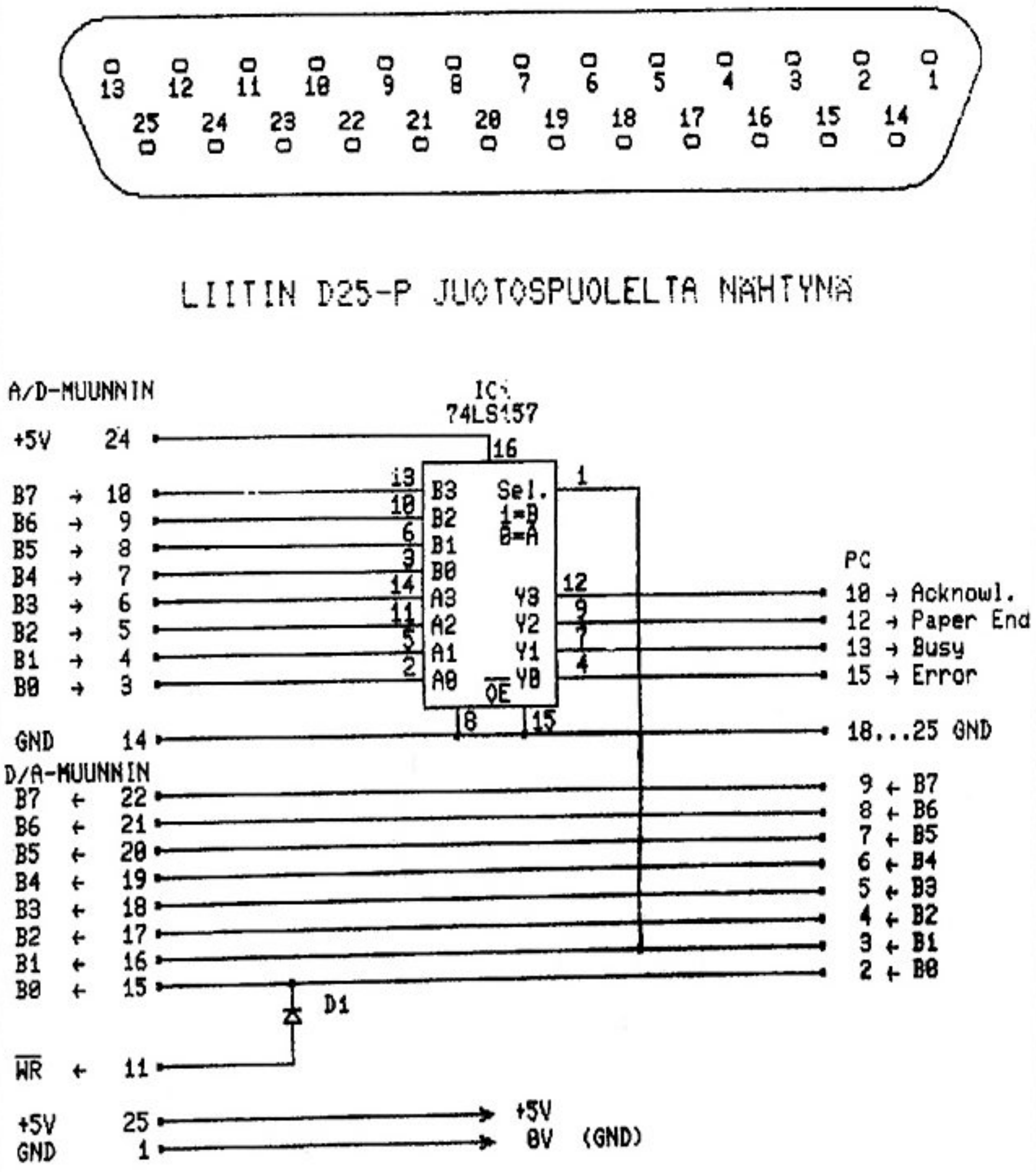
$FC:ssä olevaa lukua kasvatetaan yhdellä aina kun 256 tallennusta on suoritettu. Basiciin palataan kun on saavutettu tallennusalueen maksimiosoite.

Liitäntä PC-mikroon

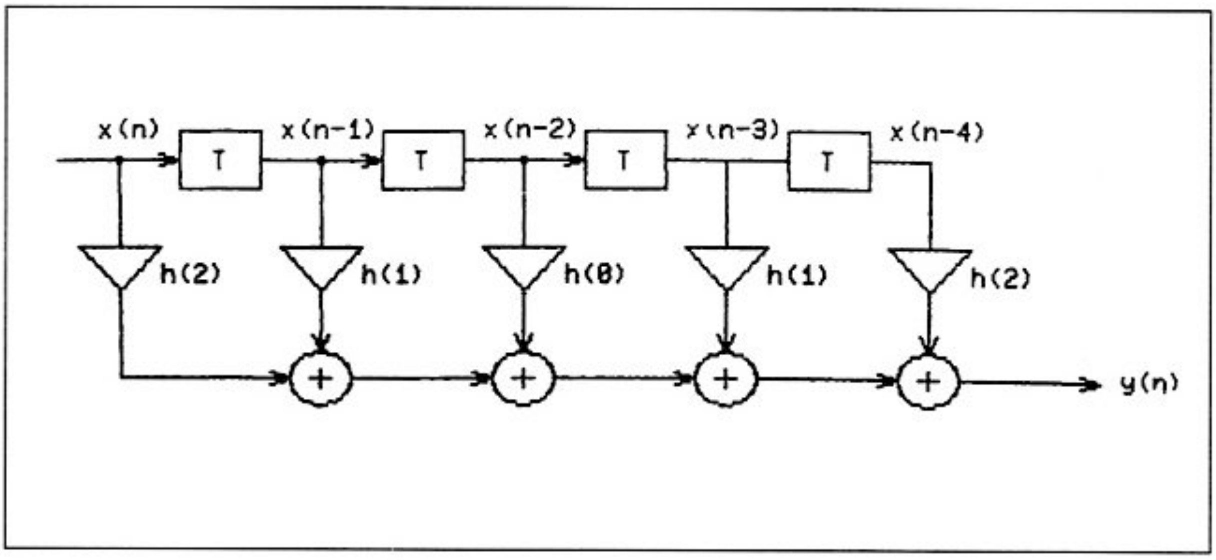
Muunnin liitetään mikron rinnakkaisporttiin. Portti sisältää 8-bittisen annon, mutta ainoastaan 5 ottobittiä. Tästä syystä joudutaan A/D-muunnoksen tulos lukemaan mikrolle kahdessa neljän bitin ryhmässä. Tätä varten on muuntimen ja mikron väliin lisätty datan valitsinpiiri (multiplekseri) 74LS157 kuvan 7 mukaisella kytkennällä.

Piirin IC1 ohjausnastan 1 ollessa ylhäällä kytkeytyvät ylemmät 4 bittiä piirin antoihin. Alemmat bitit kytkeytyvät vastaavasti nastan ollessa alhaalla. Valitsinpiirin ohjaukseen käytetään kirjoittimen dataportin bittiä 1. A/D-muuntimen käynnistykseen käytetään dataportin bittiä 0. Diodi D1 estää datalinjan oikosulkeutumisen maahan kun liittimeen 11 kytkettyä muuntimen testauspainiketta mahdollisesti painetaan.

Muunnostapahtuman aikana lähetetään muunnettu data D/A-muuntimelle tarkkailukuuntelua varten. Koska A/D-muuntimen WR-nastan on oltava muunnostapahtumen aikana ylhäällä ei alinta databittiä voida käyttää datan siirtoon.



Kuva 7. PC-lisäyksikön ja välikaapelin kytkentä.



Kuva 8. Digitaalisen suodattimen perusrakenne.

Tämä alimman bitin puuttuminen aiheuttaa sen, että äänen laatu on tarkkailu-kuuntelun aikana hiukan huonompi kuin varsinaisen toiston aikana.

Mikron rinnakkkaisliitin sisältää myös 4 kpl kirjoittimen ohjausantoja. Nämä annot ovat puskuroimattomia avoinkollektoriantoja eikä niitä hitautensa vuoksi ole voitu käyttää hyväksi tässä sovelluksessa.

A/D-muunnos ja toisto tapahtuvat mikron ajastinpiirin, 8254 tai 8253, laskurin 0 tahdistamana. Mikron nopeudella ei ole vaikutusta A/D-muunnoksen eikä toiston nopeuteen koska kaikissa PC-perheen koneissa, PC, XT, AT ja 386, on ajastinpiirin ottotaajuus sama 1,19318 MHz. Normaalitoiminnassa laskuria 0 käytetään mikron oman kellon ohjaukseen. Laskuri aiheuttaa (INT 8) keskeytyksen 18,2 kertaa sekunnissa eli aina kun siihen on tullut 65 535 pulssia.

Tällöin prosessori hakee tätä keskeytystä vastaavan suoritusosoitteen keskeytysosoitetaulukosta ja suorittaa asetetun keskeytysohjelman kasvattaen mm. muistipaikassa 40:6C olevaa vuorokaudenaikalaskuria. Keskeytys on ns. kovokeskeytys ja se tapahtuu aina pulssin tullessa riippumatta siitä, mitä prosessori on tuolloin suorittamassa. Keskeytys voidaan kuitenkin tarvittaessa estää ohjelmallisesti.

Muunninohjelmassa muutetaan keskeytystaajuus 10 kHz:ksi asettamalla laskurin 0 rekisteriin luku 119. Ajastinkeskeytys tapahtuu tällöin aina kun laskuriin on tullut 119 pulssia. Keskeytysosoitetaulukossa muutetaan keskeytysohjelman suoritusosoite haluttuun kohtaan muunninohjelmassa. Ajastinkeskeytyksen sattuessa mennään A/D-muunnoksen aikana ohjelman kohtaan, jossa luetaan muunnostulos A/D-muuntimelta, tallennetaan se muistiin ja käynnistetään uusi muunnos. Toiston aikana keskeytyksen sattuesssa luetaan muistista luku ja syötetään se ulos rinnakkaisportista D/A-muuntimelle tai ohjataan mikron kaiutinta.

Kun muunnos tai toisto on suoritettu palautetaan keskeytysvektori 8 ja keskeytystaajuus entisiksi. Mikron kellonaika korjataan oikeaksi summaamalla osoitteen 40:6C sisältöön ohjelman ajon aikana poisjääneiden kellokeskeytysten määrä.

Kaiuttimen ohjaus tapahtuu ajastinpiirin laskurin 2 avulla. Yleensä tätä laskuria käytetään erilaisten äänimerkkien tuottamiseen. Tällöin käytetään piirin toimintatilaa 3, jossa piiriltä saadaan symmetristä sakara-aaltoa. Tässä sovelluksessa käytetään sensijaan laskurin toimintatilaa 0, joka sopii erinomaisesti pulssinleveysmoduloidun, epäsymmetrisen sakara-aallon tuottamiseen.

Kun tässä toimintatilassa laskurin rekisteriin asetetaan uusi luku menee laskurin anto alas. Anto nousee ylös kun laskuriin on tullut asetettua lukua vastaava pulssimäärä 1,19 MHz:n taajuutta. Laskuriin asetetaan tässä sovelluksessa A/D-muuntimelta saatu luku jaettuna kahdella +1 eli luku väliltä 1–128.

Kaiuttimelle menevän pulssin pituus riippuu rekisteriin asetetun luvun suuruudesta. Kaiuttimen kartion keskimääräinen liike seuraa kaiuttimelle syötetyn pulssijonon keskimääräistä tasoa ja näin ollen myös alkuperäisen muunnettavan signaalin tasoa. Täten alkuperäinen signaali suodattuu korvin kuultavana esiin pulssijonosta.

Kondensaattori kaiuttimen rinnalla ja sarjavastus kaiuttimelle tulevassa johdossa parantaisi entisestään äänenlaatua ja vaimentaisi näytteenottotaajuisen (10 kHz) signaalin voimakkuutta kaiuttimella.

Ohjelmaan kuuluu myös CGA-grafiikkaa käyttävä oskilloskooppitoiminto muuntimen toiminnan tarkkailua varten. Tällä voidaan mm. todeta, että muunnin pystyy hyödyntämään koko toiminta-alueensa 0...255. Tämä näkyy oskilloskooppikuvassa siten, että suurella ottosignaalilla pitää käyrän saavuttaa ylin ja alin merkkiviiva kuvaruudulla. Mikäli näin ei tapahdu on se mahdollisesti osoitus siitä, että muuntimen ottovahvistin rajoittaa signaalia. Tällöin pitää muuntimen referenssijännitettä säätää pienemmäksi.

Myös muuntimen DC-taso voidaan säätää oskilloskooppikuvan avulla toiminta-alueen keskelle. Pienellä signaalinvoimakkuudella tulee käyrän olla keskimmäisen merkkiviivan kohdalla.

Muunnosohjelman voi tehdä kirjoittamalla listauksen 3 mukainen tekstitiedosto ja kääntämällä se jollain assemblerilla. Tällöin on ohjelmaan helppo tehdä omia muutoksia tarpeen mukaan. Toinen tapa on kirjoittaa listauksen 4 BASIC-ohjelma ja ajaa se, jolloin tuloksena saadaan valmis ADC.COM -tiedosto.

Käynnistettäessä ADC.COM -ohjelma tulee ruutuun valikko:

```
Record  Play  Save  Load  View  D/a  Quit
```

Haluttu toiminto valitaan alkukirjaimella.

R = A/D-muunnos ja tulosten tallennus muistiin

P = toisto mikron oman kaiuttimen kautta

S = muunnostulosten tallennus levylle

L = muunnostulosten lataus levyltä muistiin

V = signaalin aaltomuodon piirto kuvaruutuun

D = toisto D/A-muuntimen kautta esimerkiksi ulkoiselle vahvistimelle tai kuulokkeille

Q = poistuminen ohjelmasta

Record, Play ja D/A -toimintojen aikana kone ei reagoi näppäinten paineluun millään tavalla. Muunnosaika on ohjelman asetuksilla noin 20 sekuntia.

Save ja Load -toiminnoissa annetaan tiedoston nimi mahdollisine hakupolkuineen.

Ohjelmassa on asetettu tallennustiedoston kooksi 3*64 kilotavua. Koko on määritetty ohjelman alussa rivillä:

```
count    db        3
```

Tiedoston koon määräävä luku 3 näkyy myös Basic-listauksen rivillä 200.

Lukua voi suurentaa huoletta, sillä ohjelma huolehtii siitä, että käytettävissä olevaa muistialuetta ei ylitetä eikä dataa kirjoiteta esimerkiksi video-ram-alueelle tai muiden ohjelmien päälle.

Näytteenottotaajuuden määräävä luku 77h = 119 on myös Basic-listauksen rivillä 200. Luvun kasvattaminen hidastaa näytteenottoa, jolloin muistiin saadaan mahtumaan pitempi otos signaalista.

Muutettaessa Basic-ohjelman datarivejä pitää virheilmoituksen välttämiseksi muuttaa myös rivin lopussa olevaa tarkistussummaa.

Kuvan 7 mukaisella johdolla voidaan muunnin liittää PC, XT, AT tai 386 -mikroon. Näissä koneissa on valitettavasti vain se puute, että niissä ei yleensä ole liitintä, josta saisi ulos +5 V:n jännitteen. Tästä syystä pitää muuntimelle ottaa käyttöjännite 4,5 V:n paristosta tai muusta 4,5–5,5 V:n jännitelähteestä. Jotta mikron tai muuntimen lähtöpuskurit ja mikropiirien ottojen suojadiodit eivät kuormittuisi on hyvä tapa kytkeä muunnin mikroon vasta sitten, kun kummassakin laitteessa on jännite kytkettynä. Vastaavasti muunnin on hyvä irrottaa mik-

rosta ennen jännitteiden katkaisua. Muunnin voi olla jatkuvasti liitettynä mikroon mikäli sen käyttöjännite on otettu mikrolta.

Kuvassa 7 on esitetty liitäntäkaapelin nastajärjestys juotospuolelta nähtynä. Numerointi pätee kun välikaapelin kummassakin päässä on urosliitin (D25-P). Mikäli kaapelin muuntimen puoleiseen päähän asennetaan naarasliitin pitää johtimien paikat katsoa kuvasta eikä välittää liittimeen merkityistä numeroista.

Mikropiiri IC1 voidaan sijoittaa pienelle kytkentälevyn palalle vaikkapa muuntimen liitinkotelon sisään. Välijohdosta aiheutuvan kapasitiivisen kuormituksen vuoksi ei siitä ole syytä tehdä juurikaan 0,5 metriä pitempää.

C64-oskilloskooppi

Äänendigitointiohjelman PC-versio sisältää oskilloskooppitoiminnon, jolla digitoidun äänen signaalia voi tutkia myös kuvallisesti. Sama on mahdollista myös C64:llä listauksessa 5 esitettyä ohjelmaa käyttäen.

Ohjelma on kirjoitettu C64:lle ja Simon's Basicille, mutta se on grafiikkakomentoja ja lyhyttä konekieliosaa vaihtamalla helppo siirtää muihinkin koneisiin. Lisähienoutena PC-versioon nähden C64-ohjelma sisältää ohjelmallisen digitaalisen suodattimen, jolla signaalia voi muokata.

Tallennetun käyrän tulostuksessa sovelletaan CD-soittimissa käytettyä ylinäytteistystekniikkaa, jolla A/D-muuntimelta saatava näytemäärä voidaan moninkertaistaa tulostusvaiheessa.

Tutkittava signaali muunnetaan listauksen 5 ohjelmalla aluksi A/D-muuntimen avulla 8-bittisiksi luvuiksi, jotka tallennetaan koneen muistiin nopean konekieliohjelman avulla.

Kun käytettävissä oleva muistialue on täyttynyt, palataan Basiciin, jossa luetaan tallennettuja lukuja muistista. Tallennetun luvun suuruus riippuu suoraan alkuperäisen signaalin senhetkisestä voimakkuudesta, joten sillä määrätään kuvaruutuun piirrettävän pisteen y-koordinaatti. X-koordinaattia kasvatetaan sopiva määrä ennen seuraavan luvun lukemista muistista.

Kun seuraava luku on luettu, yhdistetään edellinen x,y-piste uuteen x,y-pisteeseen viivalla. Näin kuvaruudulle piirtyy lyhyissä pätkissä tutkittavan signaalin aaltomuotoa vastaava käyrä.

Sovellusohjelma täyttää muistia 8 kilotavua, joka vastaa tallennusaikaa 0,8 sekuntia. Käytettävän mikron muistitilan sallimissa rajoissa on tietenkin mahdollista kasvattaa tallennusaikaa.

Basic-ohjelman avulla voidaan kahlata läpi tuota muistialuetta ja tutkia tallennettua signaalia halutuista kohdista. Erityinen etu tällaisesta muistitoiminnosta tavalliseen oskilloskooppiin verrattuna on laitteen kyky tallentaa kertaluontoisia tapahtumia koneen muistiin, josta niitä voidaan jälkeenpäin tutkia kaikessa rauhassa yksityiskohtaisesti.

Suodatin

Kuva 2 esittää digitaalisen suodattimen perusrakennetta. Digitaalinen suodatin kaikessa yksinkertaisuudessaan koostuu viivepiiriketjusta (T-laatikot), jonka läpi digitoidut näytteet kulkevat. Näytteitä luetaan sisään suodattimen kellotaajuuden määräämässä tahdissa ja samanaikaisesti siirretään suodattimessa olevia näytteitä yksi askel eteenpäin. Kunkin siirron jälkeen suodattimessa olevat näytteet kerrotaan luvuilla h(0), h(1) jne. Kerrontatulosten summana saadaan suodattimen antosignaali.

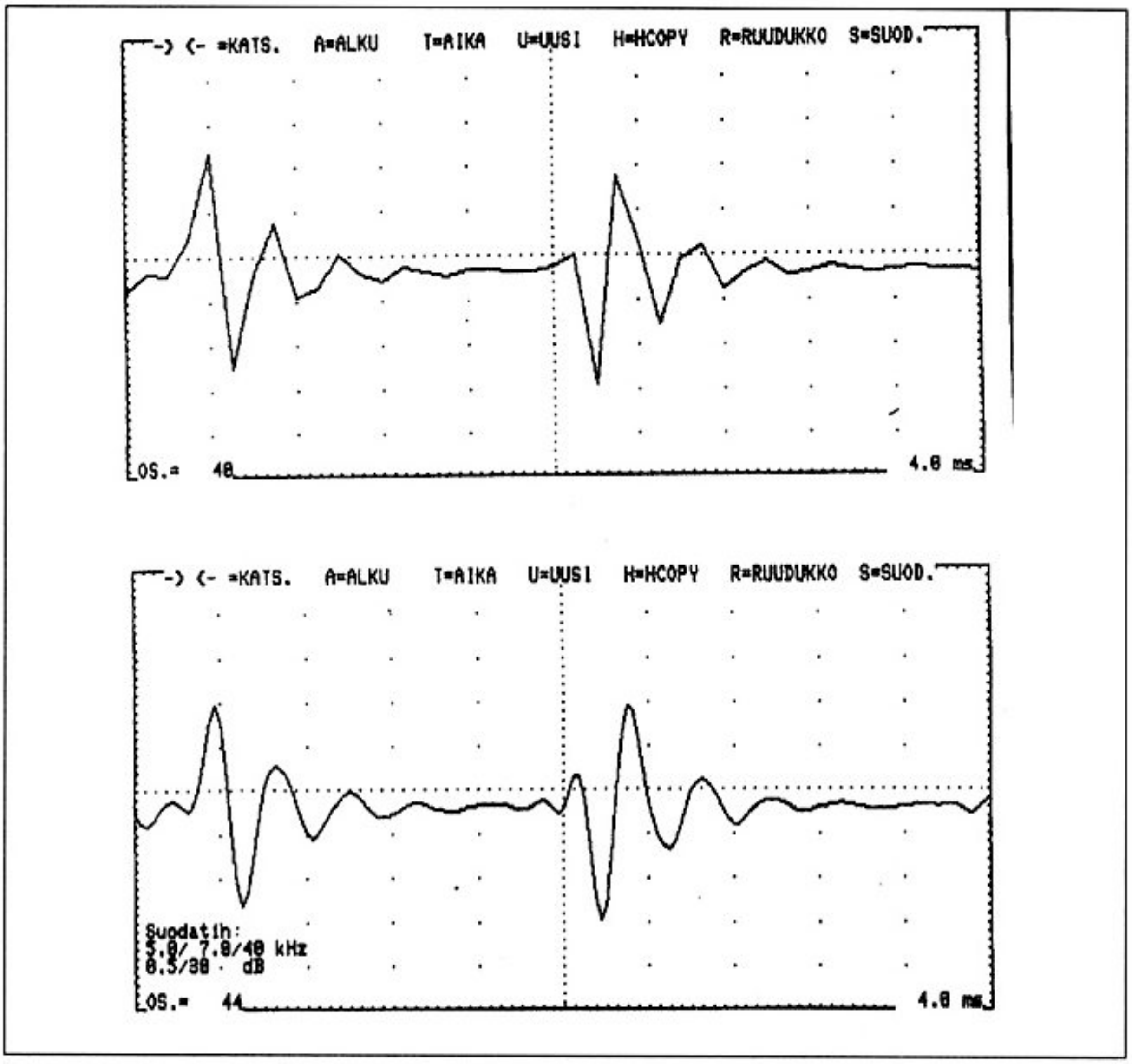
Suodattimen tyyppi (alipäästö, ylipäästö, kaistanpäästö tai kaistanesto) ja sen ominaisuudet määräytyvät kerrointen h(0), h(1), h(2) jne. mukaan. Oskilloskooppiohjelmaan on valittu alipäästö- ja kaistanpäästösuodattimien laskentamahdollisuus. Itse laskentaperiaatteeseen ei tässä yhteydessä perehdytä sen tarkemmin.

Oskilloskooppiohjelman digitaalisen alipäästösuodattimen ja kaistanpäästösuodattimen ominaisuudet on vapaasti määrättävissä ohjelmaa ajettaessa. Alipäästösuodattimen avulla voidaan tietyn taajuuden yläpuoliset signaalikomponentit poistaa tulostusvaiheessa. Kaistanpäästösuodattimella voidaan tulostaa halutun taajuusalueen signaalit.

Esimerkkinä suodattimen vaikutuksesta tulostuvaan signaaliin on kuva 9, jossa sama signaali on tulostettu ilman suodatinta sekä 5 kHz:n alipäästösuodattimen kautta. Suodattimessa on käytetty nelinkertaista ylinäytteistystä.

Muuntimen ohjaus ja muunnostulosten tallennus tapahtuu konekieliohjelman avulla. Tämä riveillä 2000–2110 oleva ohjelma on periaatteessa sama kuin edellä esitetty äänendigitointiohjelma C64:lle, mutta se on sijoitettu osoitteisiin 828–1009 ja tallennusalue on rajattu $6000–$7FFF välille.

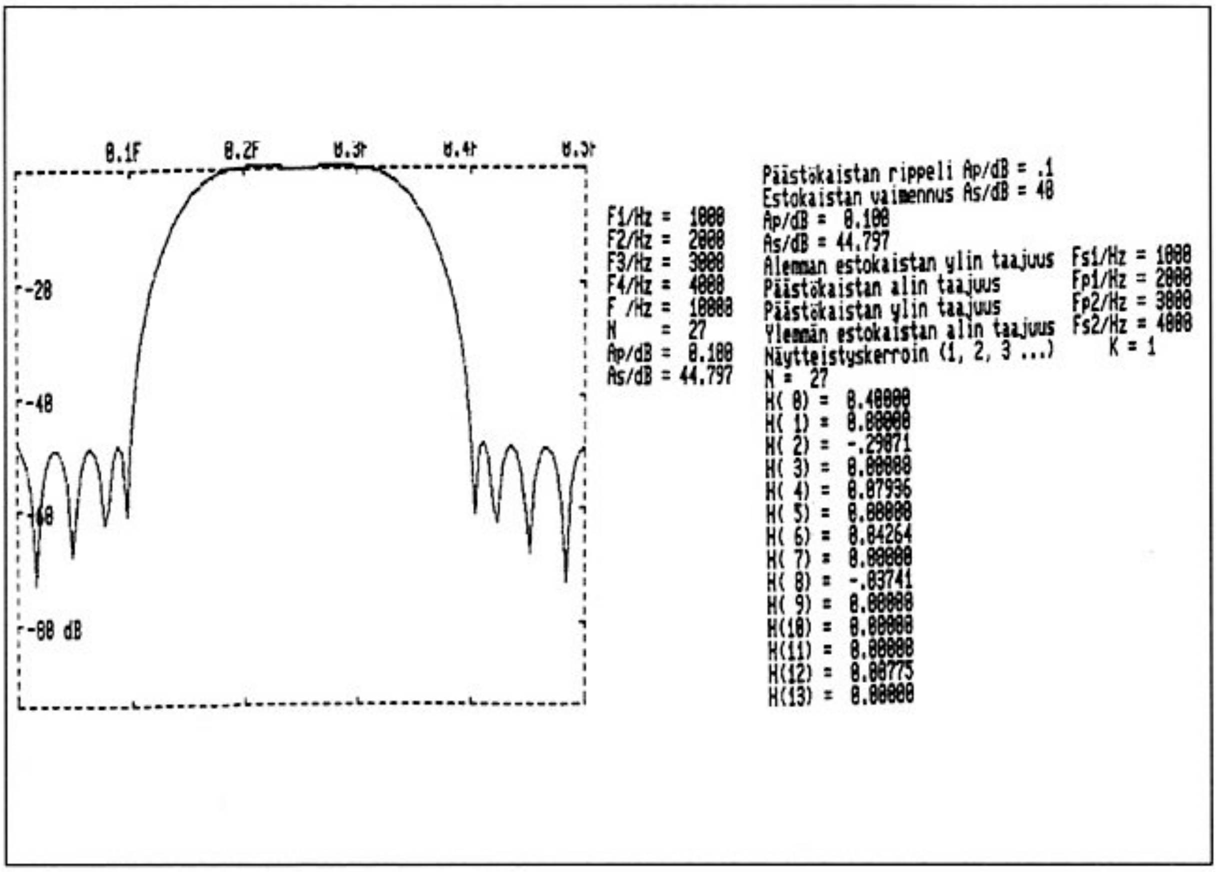
Varsinainen piirto-ohjelma on kirjoitettu Basicilla lähinnä eri koneisiin siirrettävyyden sekä toiminnan selkeyden takia. Mikäli Basic tuntuu liian hitaalta, ei ole kovinkaan vaikeaa tehdä konekieliohjelmaa, joka lukee tallennetut luvut muistista ja laskee mikä bitti on asetettava ykköseksi video-ram-alueella, jotta kuvapinnalla syttyy y-suunnassa luvun suuruutta vastaava piste. Suodatinosan kääntö konekielelle sen sijaan on jo vaikeampi tehtävä.



Kuva 9. Esimerkki suodattimen vaikutuksesta signaaliin.

Basic-ohjelman alussa ladataan konekieliohjelma data-lauseista muistiin. Tämän jälkeen käydään konekieliohjelmassa lukemassa noin 8 000 A/D-muunnostulosta muistiin. Tallennetut datat piirretään sitten kuvaruudulle. Käyrän piirto aloitetaan tallennusmuistin alusta. Kuvaruudulle piirretään noin 32 millisekunnin mittainen pätkä signaalia.

Piirto voidaan keskeyttää painamalla jotain näppäintä, jolloin ruudun yläreunaan saadaan valikko, josta päästään valitsemaan x-akselin leveys millisekunteina (T-näppäin), katselu muistissa eteenpäin (piste) tai taaksepäin (pilkku) yksi ruutu kerrallaan, paluu tallennusmuistin alkuun (A), uusi A/D-muunnos (U), digitaalinen suodatin (S) tai y-suuntainen kuvan siirto eli ns. offset (O). Offsettia tarvitaan esimerkiksi käytettäessä kaistanpäästösuodatinta, joka poistaa tasajännitekomponentin tulostettavasta signaalista.



Kuva 10. Kaistanpäästösuodattimen lähtötiedot ja niistä saadut kertoimet.

Jos valitaan x-ajaksi esimerkiksi 2 ms, piirtyy kuvaruutuun ainoastaan 20 pisteen kautta kulkeva murtoviiva. Mikäli halutaan useampia pisteitä sisältävä piirto, tulisi käyttää suodatinta.

Käytettäessä suodatinta ensimmäistä kertaa ohjelma kysyy suodattimelle arvot. Aluksi annetaan päästökaistan suurin sallittu vahvistuksen vaihtelu (rippeli) desibeleinä ja estokaistan vaimennus desibeleinä. Seuraavaksi ohjelma kysyy suodattimen taajuusrajat, joiden on kaikkien oltava alle puolet suodattimen näytteenottotaajuudesta.

Lopuksi annetaan suodattimen näytteenottokerroin, jolla A/D-muuntimen 10 kHz:n näytteenottotajuus kerrotaan. Tuloksena saadaan suodattimen näytteenottotaajuus. Kuvaruutuun tulostuvien pisteiden määrä riippuu siitä miten monikertaiseksi suodattimen näytteenottotaajuus asetetaan verrattuna 10 kHz:iin.

Tämän jälkeen kuvaruutuun tulostuu suodattimen kertaluku (N) ja lasketut kertoimet kertoimet (h(0), h(1) jne). Kertoimia tulostuu vain puolet suodattimen kertalukuun verrattuna, koska suodattimen keskimmäisen asteen kerroin on h(0) ja sen kummallakin puolella on samat kertoimet h(1), h(2) jne. Halutes-

sa voi lasketun suodattimen taajuusvastekäyrää tutkia myös ohjelman tuottaman kuvan avulla.

Kuvassa 10 on erään Sharp-mikrolla lasketun kaistanpäästösuodattimen lähtötiedot sekä niiden perusteella lasketut kertoimet ja taajuusvastekäyrä. Lukuja vertaamalla oman ohjelmasi vastaviin tulostuksiin voit todeta, että olet kirjoittanut ohjelman oikein. Luvut voivat hieman vaihdella eri koneissa näiden eri laskentatarkkuuksista johtuen. Esimerkiksi C64-version kuva ja luvut ovat hieman kuvasta 4 poikkeavia.

Kun suodattimen kertoimet on laskettu ja mahdollisesti katseltu taajuusvastekuvaa, piirtyy tutkittava signaali uudestaan ruudulle tämän suodattimen kautta vietynä.

Suodattimen hyödyn voi todeta esimerkiksi vertaamalla lähellä A/D-muuntimen rajataajuutta olevasta signaalista piirrettyjä suoraa ja suodattua kuvaa.

Lisätietoja digitaalisesta signaalinkäsittelystä löytyy mm. kirjoista:

DeFatta, Lucas, Hodgkiss :"DIGITAL SIGNAL PROCESSING", 1988.

Lynn, Fuerst :"DIGITAL SIGNAL PROCESSING WITH COMPUTER APPLICATIONS" , 1989.

Osaluettelo

Vastukset:

R1=1M

R2=100k

R3,R11,R12,R13,R14,R15,R16,R26,R27,R29,R30=10k

R4=1k10

R5,R28=1k

R7=511k

R8=390R

R9=100R

R10=220R

R18=1M5

R19=715k

R20=365k

R21=182k

R22=90k9

R23=44k2

R24=22k1

R25=11k0

R31=330R

R6=1k trimmeri

P1=100R potentiometri

Kondensaattorit:

C1=1µ polyesteri

C2=100µ tantaali

C3=100p keraaminen

C4=56p keraaminen

C5,C6,C7=100n keraaminen

C8=22µ tantaali

C9=2n2

C10=4n7

C11=47µ tantaali

Puolijohteet:

D1,D2=1N4148

D3=4V7 zeneri 400mW

D4=LED, punainen

IC2=ADC0804

IC1=TL084

IC3,IC4=4049

IC5=4066

Muut osat:

D-liitin, 25 nap., 90 ast., piirilevylle asennettava

D-liitin, 25 nap pistoke

C-64:n käyttäjän portin liitin, 24 nap.

Kiertokytkin, 4 asentoa

Painikekytkin

Kuulokeliitin

Banaani-runkoliittimiä 4 kpl

Listaus 1

Commodore 64:n äänendigitointiohjelma konekielilistauksena

```
TITLE           RECORD - Tallentaa muistiin A/D-muuntimelta
   ;saamansa datan
   ;Tämä ohjelma ohjaa C-64:n käyttäjän porttiin liitettyä
   ;AD-muunninta ja tallettaa siltä saamansa tavun mittaiset datat
   ;muistialueelle $1000 - $9FFF. Näytteenottotaajuus on n. 10 kHz
   ;ja sitä voi pienentää R_VIIVE -rivin parametria kasvattamalla.
   ;Suurempaa taajuutta ei käytetyn AD-muuntimen rajoitusten vuoksi
   ;voi käyttää. Em. muistialueen täyttyminen kestää n. 3s (10 kHz).

      ORG       $C000 ; Konekieliosa BASIC-alueen
                      ; yläpuolelle

; $C000

REC   LDA       $D011 ; Sammutetaan kuvaruutu häiriöiden
      AND       #$EF  ; vaimentamiseksi.
      STA       $D011
      LDA       #$00  ; Asetetaan CIA#2:n keskeytysmaski
      STA       $DD0D ; nollaksi.
      LDA       #$00  ; Port B input-tilaan
      STA       $DD03
      LDA       $DD02 ; Port A:n bitti 2 (PA2)
                      ; output-tilaan
      ORA       #$04
      STA       $DD02
      LDA       #$00  ; Muistialueen alarajan LowByte
      STA       $FB
      LDA       #$10  ; Muistialueen alarajan HighByte
      STA       $FC
      LDY       #$00  ; Muistin täyttö alkaa
      SEI             ; Estetaan keskeytykset

; $C025
R_SEUR          LDA   $DD00 ; PA2=0 = Aloitetaan A/D-muunnos
      AND       #$FB
      STA       $DD00
      ORA       #$04
      STA       $DD00
      JSR       R_VIIVE
      LDA       $DD01    ; Luetaan muunnoksen tulos
```

```
        STA    ($FB), Y ; ja tallennetaan se
        INY             ; Seuraava muistipaikka
        BNE    R_SEUR   ; Seuraava arvo, ellei sivu täynnä.
        INC    $FC      ; Hypätään seuraavalle sivulle.
        LDA    #$A0     ; Ollaanko jo alueen yläpäässä?
        CMP    $FC
        BNE    R_SEUR   ; Seuraava arvo jos vielä tilaa.
        LDA    $D011    ; Palautetaan kuvaruutu näkyväksi
        ORA    #$10
        STA    $D011
        CLI             ; Palautetaan keskeytykset
        RTS             ; Palataan BASIC-tulkkiin.

; $C050

R_VIIVE LDX    #$09     ; Odotetaan kunnes A/D-muunnos on  valmis
LOOP1 DEX
        BNE    LOOP1
        RTS
```

```
TITLE          PLAY - Toistaa muistiin RECORD-ohjelmalla
   ;tallennetun datan
   ;Tämä ohjelma ohjaa C-64:n käyttäjän porttiin liitettyä
   ;DA-muunninta ja toistaa RECORD-ohjelmalla alueelle $1000 -
   ;$9FFF tallennetun datan. Toiston nopeutta voi säätää P_VIIVE -
   ;rivin parametria muuttamalla. Likimain alkuperäinen nopeus
   ;saadaan arvolla $0C. DA-muuntimen antoon suositellaan tällöin
   ;liitettäväksi n. 5 kHz:n alipäästösuodin häiriöiden vaimentami-
   ;seksi.

     ORG       C060

; $C060

PLAY LDA       $D011      ; Sammutetaan kuvaruutu häiriöiden
     AND       #$EF       ; vaimentamiseksi.
     STA       $D011
     LDA       #$00       ; Asetetaan CIA#2:n keskeytysmaski
     STA       $DD0D      ; nollaksi.
     LDA       #$FF       ; Port B output-tilaan
     STA       $DD03
     LDA       $DD02      ; Port A:n bitti 2 (PA2)
                          ; output-tilaan
     ORA       #$04
     STA       $DD02
     LDA       $DD00      ; Asetetaan A/D-muuntimen anto
     AND       #$FB       ; tristate-tilaan.
     STA       $DD00
     LDA       #$00       ; Muistialueen alarajan LowByte
     STA       $FB
     LDA       #$10       ; Muistialueen alarajan HighByte
     STA       $FC
     LDY       #$00       ; Muistin täyttö alkaa
     SEI                  ; Estetaan keskeytykset
                          ; $C08D
P_SEUR         LDA ($FB), Y; Luetaan arvo muistista
     STA       $DD01      ; ja syötetään se D/A-muuntimelle
     JSR       P_VIIVE    ; Odotellaan hetki
     INY                  ; Seuraava muistipaikka
     BNE       P_SEUR     ; Seuraava arvo, ellei sivu täynnä.
     INC       $FC        ; Hypätään seuraavalle sivulle.
     LDA       #$A0       ; Ollaanko jo alueen yläpäässä?
     CMP       $FC
     BNE       P_SEUR     ; Seuraava arvo jos vielä tilaa.
     LDA       $D011      ; Palautetaan kuvaruutu näkyväksi
     ORA       #$10
     STA       $D011
```

```
     CLI                   ; Palautetaan keskeytykset
     RTS                   ; Palataan BASIC-tulkkiin.

; $C0B0

P_VIIVE LDX   #$0C         ; Viiveen pituus
LOOP2 DEX
     BNE      LOOP2
     RTS
```

Listaus 2

Commodore 64:n äänendigitointiohjelma Basic-listauksena

```
10 REM  TAMA OHJELMA LUKEE C-64:N KAYTTAJAN PORTTIIN LIITETYN AD-
11 REM  MUUNTIMEN ANTOA N. 3 SEKUNNIN AJAN  TALLETTAEN ARVOT MUIS-
12 REM  TIIN JA SYOTTAA SITTEN LUETTUA DATAA DA-MUUNTIMELLE
13 REM  YHA UUDELLEEN.

20    POKE 55,255: POKE 56,15: REM ESTETAAN BASICIN PAASY DATA-ALUEELLE
30    FOR I=49152 TO 49333: READ A: POKE I,A: NEXT I
40    SYS 49152: REM TALLENNUS
50    SYS 49248: REM TOISTO
60    GOTO 50

1000 DATA 173,017,208,041,239,141,017,208,169,000,141,013,221,169,000,141
1010 DATA 003,221,173,002,221,009,004,141,002,221,169,000,133,251,169,016
1020 DATA 133,252,160,000,120,173,000,221,041,251,141,000,221,009,004,141
1030 DATA 000,221,032,080,192,173,001,221,145,251,200,208,232,230,252,169
1040 DATA 160,197,252,208,224,173,017,208,009,016,141,017,208,088,096,000
1050 DATA 162,009,202,208,253,096,000,000,000,000,000,000,000,000,000,000
1060 DATA 173,017,208,041,239,141,017,208,169,000,141,013,221,169,255,141
1070 DATA 003,221,173,002,221,009,004,141,002,221,173,000,221,041,251,141
1080 DATA 000,221,169,000,133,251,169,016,133,252,160,000,120,177,251,141
1090 DATA 001,221,032,176,192,200,208,245,230,252,169,160,197,252,208,237
1100 DATA 173,017,208,009,016,141,017,208,088,096,000,000,000,000,000,000
1110 DATA 162,012,202,208,253,096
```

Listaus 3

A/D-muuntimen äänendigitointiohjelma PC:lle.

```
; A/D-muuntimen ohjausohjelma PC:lle
; Liitäntä rinnakkaisporttiin multiplekserin 74LS157
; kautta

prin macro      teksti
; Makro, joka tulostaa $:iin päättyvän tekstin
      mov       dx, offset teksti
      mov       ah, 9
      int       21h
      endm

cseg segment
      assume    cs:cseg, ds:cseg, es:cseg, ss:cseg
      org       100h
start:          jmp       asetukset
count db        3
counter db      0
laskuri db      119   ; 10 kHz:n keskeytystaajuus
komm1 db        0ah, 0dh,
      db        'Record    Play  Save  LoadViewD/aQuit',0ah,0dh,'$'
komm2 db        0ah, 0dh, 'File: $'
komm3 db        0ah, 0dh, '-    Quit$'
komm4 db        0ah, 0dh, '??', 0ah, 0dh, '$'

asetukset:
      mov       bx, 0ffffh  ; yritetään varata ffffh*10h muistia
      mov       ah, 4ah    ; varaus epäonnistuu, jolloin
      int       21h        ; bx = vapaan muistin määrän /10h
      mov       ax, offset pino
      mov       sp, ax     ; pinon asetus
      mov       cl, 4
      shr       ax, cl     ; ohjelman+pinon pituus jaetaan 10h:lla
      inc       ax         ; pyöristys ylöspäin
      sub       bx, ax     ; vähenn. ohjelman pituus vapaasta muistista
      mov       dx,cs
      add       dx,ax      ; summataan cs:ään ohjelman pituus/10h+1
      mov       tallseg, dx ; saadaan talletuksen aloitussegmentti
      rol       bx, cl     ; bx:n ylin puolitavu sen alimmaksi
      and       bl, 0fh    ; 64 k:n lohkojen määrä vapaassa muistissa
      jz        ulos       ; muistia alle 64 k - poistuminen ohjelmasta
      cmp       bl, count
```

```
        jae     prt           ; jos tulos pienempi kuin count, niin
        mov     count,bl      ; asetetaan luku count:in arvoksi
prt:
        mov     ax, 40h       ; haetaan kirjoitinportin osoite
        mov     es,ax         ; osoitteesta 40:08
        mov     ax, es:[8]
        mov     pr_portti,  ax
start1:
        print   komm1         ; tekstirivin tulostus ruutuun
        mov     al,   count
        mov     counter,al    ;lohkolaskurin asetus
luenapp:
        mov     ah, 1
        int     21h           ; näppäimen luenta al:ään
        and     al, 5fh       ; muutetaan isoksi kirjaimeksi
        cmp     al, 'Q'
        je      ulos
        cmp     al, 'R'
        je      muunnos
        cmp     al, 'P'
        je      toisto
        cmp     al, 'D'
        je      toisto
        cmp     al, 'S'
        je      talleta
        cmp     al, 'L'
        je      lataa
        cmp     al, 'V'
        je      skpiin
        jmp     luenapp
skpiin:
        call    skooppi
        jmp     start1
talleta:
        call    save
        jmp     start1
lataa:
        call    load
        jmp     start1
ulos:
        mov     ax, 4c00h     ; poistuminen DOS:iin
        int     21h
muunnos:
        mov     dx, offset rec ; uusi keskeytysosoite dx:ään
        call    aseta_vektori
```

```
      mov     cl, 4           ; luku bittien pyöritystä varten
      mov     dx, pr_portti   ; kirjoittimen dataportin osoite dx:ään
      jmp     asr
toisto:
      push    ax
      mov     dx, offset play
      call    aseta_vektori
      mov     dx, pr_portti
      xor     bh, bh    ; bh = 0, toisto d/a-muuntimella
      pop     ax
      cmp     al, 'D'   ; onko valittu d/a-muuntimella toisto
      jz      asr
      mov     dx, 42h   ; laskurin 2 osoite dx:ään
      mov     al, 10010000b
      out     43h, al   ; laskurin 2 toimintamuodon asetus
      in      al, 61h
      or      al, 3     ; kaiutin ja laskuri päälle
      out     61h, al
      inc     bh        ; bh0,merkiksi,että toisto kaiuttimelle
asr:
      mov     ax, tallseg ; talletussegm.
asreg:
      mov     es, ax
      mov     si, 1     ; talletusosoite
      sti               ; kesk. sallitaan

testi:
      or      si, si    ; tässä odotellaan ajastinkeskeytyksiä
      jnz     testi
      cli               ; kesk. esto
      mov     ax, es
      add     ax, 1000h   ; segmentin kasvatus
      dec     counter
      jnz     asreg
      call    palauta_vektori
      jmp     start1
```

```
; --- Keskeytysaliohjelmat:

; --- A/D-muuntimen ohjaus ja luenta
rec:
      xor     al, al    ; bitti 0 eli wr -nasta alas hetkeksi,
      out     dx, al    ; uusi muunnos käynnistyy
      mov     al, 00000011b  ; multiplekserin ohjausbitti (1) ja
      out     dx, al    ; muuntimen wr-bitti (0) ylös
      inc     dx        ; kirjoittimen statusportin osoite dx:ään
      in      al, dx    ; luetaan A/D-muunnoksen 4 ylintä bittiä
      shl     al, 1     ; ja siirretään ne ah -rekisteriin
      shl     ax, cl
      dec     dx        ; dataportin osoite dx:ään
      mov     al, 00000001b  ; multiplekserin ohjausbitti (1) alas
      out     dx, al
      inc     dx        ; statusportin osoite dx:ään
      in      al, dx    ; luetaan A/D-muunnoksen 4 alinta bittiä
      shl     al, 1     ; ja siirretään ne ah:n alimmiksi biteiksi
      shl     ax, cl    ; edelliset 4 bittiä siirtyvät ylimmiksi
      dec     dx        ; dataportin osoite dx:ään
      mov     al, ah    ; muunnostulos al:ään
      or      al, 1     ; al:n alin bitti (wr) ykköseksi
      out     dx, al    ; muunnostulos D/A-muuntimelle
      mov     es:[si], ah ; talletus muistiin
      inc     si        ; talletusosoitteen kasvatus
      mov     al, 20h
      out     20h, al     ; sallitaan uusi keskeytys
      iret

; --- Toisto kaiuttimelle tai D/A-muuntimelle

play:
      mov     al, bl    ; ed. luettu data bl:stä
      out     dx, al    ; laskurin asetus tai data ulos
      mov     bl, es:[si] ; luetaan data bl:ään
      inc     si
      or      bh, bh
      jz      poisr     ; toisto d/a-muuntimelle
      shr     bl, 1     ; jako 2:lla
      inc     bl        ; varmistetaan,että aina bl = 1
poisr:
      mov     al, 20h
      out     20h, al
      iret
; --- Muut aliohjelmat:
```

```
; --- Keskeytysvektorin asetus

aseta_vektori:
     cli                  ; kesk. esto
     mov     ax, 3508h    ; haetaan nykyinen INT8:n osoite
     int     21h
     mov     word ptr vanhaosoite, bx; ja talletetaan se
     mov     word ptr vanhaosoite+2, es
     mov     ax, 2508h
     int     21h       ; asetetaan oma keskeytysosoite dx:stä
     mov     al, 11111110b  ; vain ajastinkeskeytykset (0) sallitaan
     out     21h, al
     mov     ah, laskuri

; --- Keskeytystaajuuden asetus

k_aika:
     mov     al, 34h
     out     43h, al      ; asetetaan laskurin 0 toimintatila
     mov     al, ah
     out     40h, al      ; asetetaan taajuusjakajan alempi tavu
     xor     al, al
     out     40h, al      ;  ja ylempi tavu
     ret

; --- Keskeytysvektorin palautus entiseksi

palauta_vektori:
     mov     al, 0ffh
     out     21h, al      ; kesk. esto
     push    ds
     lds     dx, vanhaosoite
     mov     ax, 2508h
     int     21h       ; INT8:n osoitteen palautus
     pop     ds

; --- Kellonajan korjaus

     mov     ax, 40h
     mov     es, ax       ; time-of-day -laskurin segmentti
     mov     bx, 6ch      ; ja osoite
     mov     cl, laskuri
     mov     al, count    ; kertolaskun tuloksena saadaan
     mul     cl           ; kulunut aika (1/18,2-sekunteina)
     add     es:[bx], ax ; summataan time-of-day -laskuriin
```

```
     adc     word ptr es:[bx+2], 0
     cmp     word ptr es:[bx+2], 18h ; onko vuorokausi vaihtunut ?
     jb      asetettu
     ja      ylitys
     cmp     word ptr es:[bx], 0bh
     jb      asetettu
ylitys:
     sub     word ptr es:[bx], 0bh
     sbb     word ptr es:[bx+2], 18h
asetettu:
     in      al, 61h
     and     al, 11111100b
     out     61h, al   ; kaiutin pois päältä
     mov     ah, 0     ; keskeytysajastimen palautus normaaliksi
     call    k_aika
     xor     al, al    ; sallitaan kaikki keskeytykset
     out     21h, al
     sti
     ret

; --- Tiedoston talletus levylle

save:
     call    anna_nimi
     mov     ah, 3ch
     mov     cx, 0     ; attribuutti
     int     21h       ; luodaan ja avataan tiedosto
     jc      virhe
     mov     bx, ax    ; ax = handle,  talteen bx:ään
     mov     ax, tallseg
tall:
     mov     ds, ax
     xor     dx, dx    ; ds:dx osoittaa datan alkuun
     mov     ah, 40h
     mov     cx, 0ffffh  ; tavumäärä
     int     21h
     jc      virhe_1
     mov     ax, ds
     add     ax, 1000h
     dec     cs:counter  ; 64 k:n lohkojen laskenta
     jnz     tall
     jmp     close
```

```
; --- Tiedoston luenta levyltä muistiin

load:
      call     anna_nimi
      mov      ax, 3d00h
      int      21h      ; tiedoston avaus
      jc       virhe
      mov      bx, ax
      mov      ax, tallseg
lue:
      mov      ds, ax
      mov      cx, 0ffffh  ; tavumäärä
      xor      dx, dx   ; ds:dx data bufferin alku
      mov      ah, 3fh
      int      21h
      jc       virhe_1
      or       ax, ax   ; onko tiedosto loppu (ax = 0)
      jz       close    ; tiedosto suljetaan
      mov      ax, ds
      add      ax, 1000h   ; segmentin kasvatus
      dec      cs:counter
      jnz      lue
close:
      push     cs
      pop      ds       ; datasegmentinpalautus
      mov      ah, 3eh
      int      21h
      ret
virhe:
      push     cs
      pop      ds
      print    komm4    ; virheilmoituksen tulostus
      ret
virhe_1:
      call     virhe
      jmp      close
anna_nimi:
      print    komm2
      lea      dx, nimi  ; ds:dx osoittaa nimen paikkaa
      mov      bx, dx
      mov      byte ptr [bx], 30 ; nimen alkuun sen max. pituus (30)
      mov      ah, 0ah
      int      21h      ; teksti näppäimistöltä
      xor      ah, ah
      mov      al, [bx]+1  ; luetaan annetun tekstin pituus ja
```

```
     add     bx, ax   ; summataan tekstin alkuosoitteeseen
     mov     byte ptr [bx]+2, 0   ; lisätään loppuun 0
     lea     dx, nimi+2  ; aset. dx osoittamaan tekstin alkukohtaa
     ret

; --- Piirretään signaalin käyräkuva ruutuun

skooppi:
     mov     ah, 0fh
     int     10h          ; näytön mode al:ään
     mov     modenum, al ; ja talletetaan
skooppi1:
     mov     si, 0        ; muistiosoite
     mov     es, tallseg ; muistisegm.
eteenp:
     mov     ax, 4      ; mode 4 = 320x200 pistettä,4 väriä (CGA)
     int     10h        ; samalla ruudun tyhjennys
     mov     cx, 315     ; cx = pisteen x-koordinaatti

asteik:
     mov     dx, 128     ; dx = pisteen y-koordinaatti
ast:
     call    piste      ; asteikkoviivojen piirto
     sub     dx, 64
     jns     ast
     sub     cx, 5
     jnz     asteik
     mov     di, 64
luem:
     xor     dh, dh
     mov     dl, es:[si] ; luetaan A/D-muunnoksen tulos
     inc     si          ; luentaosoitteen kasvatus
     shr     dl, 1       ; muunnostuloksen jako 2:lla
     xchg    dx, di      ; uusi y-piste di:hin,edellinen dx:ään
     jmp     valm
viiva:
     dec     dx
     jmp     valm
viiva1:
     inc     dx          ; viivan piirto ed. y-pisteeseen
valm:
     call    piste
     cmp     dx, di
     jb      viiva1
     ja      viiva
```

```
ohi:
     inc      cx        ; x-koordinaatin kasvatus
     cmp      cx, 320
     jne      luem
     print    komm3
getkey:
     mov      ah, 7     ; ohjaustieto näppäimistöltä
     int      21h
     and      al, 5fh   ; muutetaan isoiksi kirjaimiksi
     cmp      al, 'Q'
     je       poistu
     or       al, al
     jnz      getkey
     int      21h       ; luetaan laajennettu koodi
     cmp      al, 77    ; -
     je       eteenp
     cmp      al, 75    ;
     je       taakse
     jmp      getkey
taakse:
     sub      si, 640    ; muistissa 640 merkkiä taaksepäin
     jmp      eteenp
poistu:
     mov      ah, 0
     mov      al, modenum ; palautetaan alkuperäinen näytön tila
     int      10h
     ret
piste:
     mov      al, 1     ; al = palettikoodi (väri)
     mov      bh, 0     ; bh = sivu
     mov      ah, 0ch    ; pisteen paikka: cx = x,dx = y
     int      10h       ; pisteen sytytys
     ret
     even
loppu equ $             ; alustamaton data-alue
vanhaosoite   equ dword ptr loppu
tallseg       equ word ptr vanhaosoite+4
pr_portti     equ word ptr tallseg+2
modenum       equ byte ptr pr_portti+2
mrk           equ byte ptr modenum+1
nimi  equ byte ptr mrk+1
pino  equ word ptr loppu+100h
cseg  ends
     end      start
```

Listaus 4

A/D-muuntimen äänendigitointiohjelma PC:lle Basic-listauksena

```
100 REM         OHJELMA, JOKA LUO TIEDOSTON ADC.COM
110 PRINT "Tiedoston luonti ...":OPEN "ADC.COM" AS #1 LEN=1
120 FIELD #1,1 AS D$
130 FOR I=200 TO 660 STEP 10:FOR J=1 TO 16:READ A$
140 D=VAL("&h"+A$):TS=TS+D:LSET D$=CHR$(D):PUT #1:NEXT J
150 READ S$:IF VAL("&h"+S$)TS THEN 170
160 TS=0:NEXT I:CLOSE:END
170 PRINT "Virhe rivillä "; I:END
200 DATA EB, 51, 90,3,0, 77,A,D, 52, 65, 63, 6F, 72, 64, 20, 20,4FC
210 DATA 50, 6C, 61, 79, 20, 20, 53, 61, 76, 65, 20, 20, 4C, 6F, 61, 64,525
220 DATA 20, 20, 56, 69, 65, 77, 20, 20, 44, 2F, 61, 20, 20, 51, 75, 69,45E
230 DATA 74,A,D, 24,A,D, 46, 69, 6C, 65, 3A, 20, 24,A,D, 2D,308
240 DATA 3E, 20, 20, 3C, 2D, 20, 20, 51, 75, 69, 74, 24,A,D, 3F, 3F,383
250 DATA A,D, 24, BB, FF, FF, B4, 4A, CD, 21, B8, E8,4, 8B, E0, B1,8A0
260 DATA 4, D3, E8, 40, 2B, D8, 8C, CA,3, D0, 89, 16, EC,3, D3, C3,84F
270 DATA 80, E3,F, 74, 56, 3A, 1E,3,1, 73,4, 88, 1E,3,1, B8,471
280 DATA 40,0, 8E, C0, 26, A1,8,0, A3, EE,3, BA,6,1, B4,9,56F
290 DATA CD, 21, A0,3,1, A2,4,1, B4,1, CD, 21, 24, 5F, 3C, 51,4EC
300 DATA 74, 29, 3C, 52, 74, 2A, 3C, 50, 74, 35, 3C, 44, 74, 31, 3C, 53,4B2
310 DATA 74,F, 3C, 4C, 74, 10, 3C, 56, 74,2, EB, DC, E8, A5,1, EB,6D7
320 DATA CA, E8,D,1, EB, C5, E8, 35,1, EB, C0, B8,0, 4C, CD, 21,82B
330 DATA BA, 1F,2, E8, 82,0, B1,4, 8B, 16, EE,3, EB, 22, 90, 50,679
340 DATA BA, 44,2, E8, 72,0, 8B, 16, EE,3, 32, FF, 58, 3C, 44, 74,669
350 DATA F, BA, 42,0, B0, 90, E6, 43, E4, 61,C,3, E6, 61, FE, C7,7D4
360 DATA A1, EC,3, 8E, C0, BE,1,0, FB,B, F6, 75, FC, FA, 8C, C0,950
370 DATA 5,0, 10, FE,E,4,1, 75, EA, E8, 64,0, E9, 6C, FF, 32,657
380 DATA C0, EE, B0,3, EE, 42, EC, D0, E0, D3, E0, 4A, B0,1, EE, 42,A0B
390 DATA EC, D0, E0, D3, E0, 4A, 8A, C4,C,1, EE, 26, 88, 24, 46, B0,8AA
400 DATA 20, E6, 20, CF, 8A, C3, EE, 26, 8A, 1C, 46,A, FF, 74,4, D0,793
410 DATA EB, FE, C3, B0, 20, E6, 20, CF, FA, B8,8, 35, CD, 21, 89, 1E,8D5
420 DATA E8,3, 8C,6, EA,3, B8,8, 25, CD, 21, B0, FE, E6, 21, 8A,77C
430 DATA 26,5,1, B0, 34, E6, 43, 8A, C4, E6, 40, 32, C0, E6, 40, C3,788
440 DATA B0, FF, E6, 21, 1E, C5, 16, E8,3, B8,8, 25, CD, 21, 1F, B8,744
450 DATA 40,0, 8E, C0, BB, 6C,0, 8A,E,5,1, A0,3,1, F6, E1,5CE
460 DATA 26,1,7, 26, 83, 57,2,0, 26, 83, 7F,2, 18, 72, 11, 77,36C
470 DATA 6, 26, 83, 3F,B, 72,9, 26, 83, 2F,B, 26, 83, 5F,2, 18,379
480 DATA E4, 61, 24, FC, E6, 61, B4,0, E8, A8, FF, 32, C0, E6, 21, FB,9E3
490 DATA C3, E8, 6C,0, B4, 3C, B9,0,0, CD, 21, 72, 54, 8B, D8, A1,778
500 DATA EC,3, 8E, D8, 33, D2, B4, 40, B9, FF, FF, CD, 21, 72, 4C, 8C, 93D
```

```
510 DATA D8,5,0, 10, 2E, FE,E,4,1, 75, E7, EB, 2D, 90, E8, 3F,657
520 DATA 0, B8,0, 3D, CD, 21, 72, 29, 8B, D8, A1, EC,3, 8E, D8, B9,790
530 DATA FF, FF, 33, D2, B4, 3F, CD, 21, 72, 21,B, C0, 74,C, 8C, D8,826
540 DATA 5,0, 10, 2E, FE,E,4,1, 75, E3,E, 1F, B4, 3E, CD, 21,4B9
550 DATA C3,E, 1F, BA, 4C,1, B4,9, CD, 21, C3, E8, F3, FF, EB, EA,914
560 DATA BA, 34,1, B4,9, CD, 21, 8D, 16, F2,3, 8B, DA, C6,7, 1E,682
570 DATA B4,A, CD, 21, 32, E4, 8A, 47,1,3, D8, C6, 47,2,0, 8D,60B
580 DATA 16, F4,3, C3, B4,F, CD, 10, A2, F0,3, BE,0,0, 8E,6,657
590 DATA EC,3, B8,4,0, CD, 10, B9, 3B,1, BA, 80,0, E8, 5E,0,5FD
600 DATA 83, EA, 40, 79, F8, 83, E9,5, 75, F0, BF, 40,0, 32, F6, 26,841
610 DATA 8A, 14, 46, D0, EA, 87, D7, EB,6, 90, 4A, EB,2, 90, 42, E8,86E
620 DATA 3C,0, 3B, D7, 72, F8, 77, F2, 41, 81, F9, 40,1, 75, DE, BA,82A
630 DATA 3D,1, B4,9, CD, 21, B4,7, CD, 21, 24, 5F, 3C, 51, 74, 16,52C
640 DATA A, C0, 75, F2, CD, 21, 3C, 4D, 74, A8, 3C, 4B, 74,2, EB, E6,792
650 DATA 81, EE, 80,2, EB, 9C, B4,0, A0, F0,3, CD, 10, C3, B0,1,810
660 DATA B7,0, B4,C, CD, 10, C3,0,0,0,0,0,0,0,0,0,317
```

Listaus 5

Oskilloskooppiohjelma C64:lle

```
Listauksessa on käytetty seuraavia erikoismerkintöjä:

     Ruudun tyhjennys (SHIFT-HOME)
     <MI>      Pii (SHIFT-POTENSSIINKOROTUSNUOLI)
     <L><R>    A      Control-A (CTRL-A)
```

```
0    REM ==============================================
1    REM OSKILLOSKOOPPI C-64:LLE JA SIMON'S BASICILLE
2    REM ==============================================

5    REM ALUSTUKSET
6    REM ==========
10   POKE 55,255: POKE 56,95: REM DATA-ALUEELLE 6000-7FFF
20   DIM HD(200), AK(200), H(200), A(200), K(20)
30   FOR I=828 TO 1009: READ A: POKE I,A: NEXT
40   HIRES 1,0

100  REM AD-MUUNNOS
110  REM ==========
120  SYS 828: REM MUUNNETAAN 8 KT DATAA
130  T=32: S=0
140  A1=24576
150  GOTO 500: REM PIIRRETAAN DATA

160  REM VALIKKO
170  REM =======
180  TEXT 0,0,"<L><R>  A  A=ALKU T=AIKA U=UUSI S=SUOD O=OFFS.",1,1,8
190  TEXT 0,191,"<L><R>   AOSOITE"+STR$(A1-24576),1,1,8
195  TEXT 160,191,"<L><R> AAIKA "+STR$(T)+" MS",1,1,8
200  GET A$: IF A$="" THEN 200
210  IF A$="A" THEN A1=24576: GOTO 500
220  IF A$="U" THEN 100
230  IF A$="T" THEN 300
240  IF A$="S" THEN 700
250  IF A$="O" THEN 350
260  IF A$="." THEN A1=A1+320*DT: GOTO 500
270  IF A$="," THEN A1=A1-320*DT: GOTO 500
290  GOTO 200

300  REM UUSI AIKA
```

```
310   REM =========
320   NRM: PRINT"<L><R>AIKA ON NYT (MS): "; T
330   INPUT "ANNA UUSI AIKA:"; T
340   GOTO 500

350   REM UUSI OFFSET
360   REM ===========
370   NRM: PRINT"<L><R>OFFSET ON NYT : "; OF
380   INPUT "ANNA UUSI OFFSET :"; OF
390   GOTO 500

500   REM PIIRTO
510   REM ======
520   A0=A1: DT=T/32: DX=1
530   IF T  THEN DX=32/T: DT=1
540   HIRES 1,0: X0=0: X=0
550   Y0=PEEK(A0)-OF
560   IF S=1 THEN 1340
570   X=X+DX: IF X320 OR A032766 THEN 160
580   A0=A0+DT
590   Y=PEEK(A0)-OF
600   IF Y THEN Y=0
610   IF Y0 THEN Y0=0
620   IF Y0199 THEN Y0=199
630   IF Y199 THEN Y=199
640   LINE X0,Y0,X,Y,1
650   X0=X: Y0=Y
660   GET A$: IF A$"" THEN 160
670   GOTO 570

700   REM FIR-SUODIN
710   REM ==========
720   NRM: PRINT "<L><R>";
730   IF H(0)=0 THEN 770
740   PRINT "UUDET SUODATINARVOT (K/E)?"
750   GET A$: IF A$="" THEN 750
760   IF A$="E" THEN 1160
770   K(1)=1: PRINT"<L><R>";
780   FOR I=2 TO 20: K(I)=K(I-1): NEXT
790   INPUT "1=ALIPAASTO,2=KAISTANPAASTO "; SU
800   INPUT "PAASTOKAISTAN RIPPELI AP/DB "; AP
810   INPUT "ESTOKAISTAN VAIMENN.  AS/DB "; AS
820   RS=10^-(.05*AS)
830   RP=(10^(.05*AP)-1)/(10^(.05*AP)+1)
840   IF RS<R> THEN RM=RS: ELSE: RM=RP
```

```
850  Q1=20*LOG((1+RM)/(1-RM))/LOG(10): PRINT "AP/DB="; Q1
860  W1=-20*LOG(RM)/LOG(10): PRINT "AS/DB="; W1
870  IF W1<M=>21 THEN D=.9222: ELSE: D=(W1-7.95)/14.36
880  ON SU GOSUB 1500, 1600
890  INPUT "YLINAYTTEISTYSKERROIN  K "; KF
900  F=KF*10000
910  N=INT(F*D/DF+2): IF INT(N/2)=N/2 THEN N=N+1
920  PRINT "N="; N
930  IF W1<M=>21 THEN AL=0
940  IF W121 AND W1<M=>50 THEN AL=.5842*(W1-21)^.4+.07886*(W1-21)
950  IF W150 THEN AL=.1102*(W1-8.7)
960  IA=1
970  FOR I=1 TO 10: IA=IA+((.25*AL*AL)^I)/(K(I)^2): NEXT
980  ON SU GOTO 990, 1000
990  HD(0)=2*FC/F: GOTO 1010
1000 HD(0)=2/F*(C2-C1)
1010 I=0
1020 GOSUB 1100
1030 FOR I=1 TO (N-1)/2
1040 ON SU GOTO 1050,1060
1050 HD(I)=(2*FC/F)*SIN(2*<MI>*I*FC/F)/(2**I*FC/F): GOTO 1070
1060 HD(I)=(1/(I*<MI>))*(SIN(2**I*C2/F)-SIN(2**I*C1/F))
1070 GOSUB 1100
1080 NEXT
1090 S=1: GOTO 1160

1100 BE=AL*(1-(2*I/(N-1))^2)^.5
1110 IB=1
1120 FOR J=1 TO 10: IB=IB+((.25*BE^2)^J)/(K(J)^2): NEXT
1130 AK(I)=IB/IA: H(I)=HD(I)*AK(I)
1140 PRINT "H("; I; ")="; H(I)
1150 RETURN

1160 REM TAAJUUSVASTE
1170 REM ============
1180 PRINT "TAAJUUSVASTEEN KUVA (K/E) ?"
1190 GET A$: IF A$="" THEN 1190
1195 IF A$"K" THEN 1330
1200 HIRES 1,0: X0=0: X1=0: TM=1/F
1210 FOR FX=F/160 TO F/2 STEP F/160
1220 H=H(0)
1230 FOR I=1 TO (N-1)/2
1240 H=H+2*H(I)*COS(2*<MI>*I*FX*TM)
1250 NEXT
1260 H=ABS(H)
```

```
1270 Y=LOG(H)*20/LOG(10)
1280 X1=X1+4: Y1=10-Y: IF X1=4 THEN Y0=Y1
1285 IF Y1 THEN Y1=0
1286 IF Y0 THEN Y0=0
1287 IF Y0199 THEN Y0=199
1288 IF Y1199 THEN Y1=199
1290 LINE X1,Y1,X0,Y0,1
1300 Y0=Y1: X0=X1
1310 NEXT
1320 GET A$: IF A$="" THEN 1320
1330 S=1

1340 REM INTERPOLOINTI
1345 REM =============
1350 HIRES 1,0
1355 DT=10000/F: DX=DT*32/T: A0=A1
1360 Y0=100: X0=0: X=0
1365 FOR I=N-1 TO 0 STEP-1
1370 A(I)=PEEK(A0): A0=A0+DT
1375 NEXT
1380 X=X+DX: IF X320 OR A032766 THEN 160
1385 GET A$: IF A$"" THEN 160
1390 A(0)=PEEK(A0): A0=A0+DT
1395 FOR I=N-1 TO 1 STEP -1
1400 A(I)=A(I-1)
1405 NEXT
1410 Y=A((N-1)/2)*H(0)
1415 FOR J=0 TO (N-1)/2-1
1420 Y=Y+(A(J)+A(N-1-J))*H((N-1)/2-J)
1421 NEXT: Y=Y-OF
1425 IF Y THEN Y=0
1426 IF Y0 THEN Y0=0
1427 IF Y0199 THEN Y0=199
1428 IF Y199 THEN Y=199
1430 LINE X0,Y0,X,Y,1
1435 X0=X: Y0=Y
1440 GOTO 1380

1500 INPUT "PAASTOKAISTAN YLIN T. FP/HZ "; F1
1510 INPUT "ESTOKAISTAN ALIN T.  FS/HZ "; F2
1520 DF=F2-F1: FC=(F1+F2)/2
1530 RETURN

1600 INPUT "ALEMM. ESTOK.YLIN T. FS1/HZ "; F1
1610 INPUT "PAASTOK. ALIN T.  FP1/HZ "; F2
```

```
1620 INPUT "PAASTOK. YLIN T.  FP2/HZ "; F3
1630 INPUT "YLEMM. ESTOK.ALIN T. FS2/HZ "; F4
1640 D1=F2-F1: D2=F4-F3: IF D1D2 THEN DF=D2: ELSE: DF=D1
1650 C1=F2-DF/2: C2=F3+DF/2
1660 RETURN

1990 REM KASETTIPUSKURIIN SIJOITETTAVA KONEKIELIOHJELMA
2000 DATA 173,017,208,041,239,141,017,208,169,000,141,013,221,169,000,141
2010 DATA 003,221,173,002,221,009,004,141,002,221,169,000,133,251,169,096
2020 DATA 133,252,160,000,120,173,000,221,041,251,141,000,221,009,004,141
2030 DATA 000,221,032,140,003,173,001,221,145,251,200,208,232,230,252,169
2040 DATA 128,197,252,208,224,173,017,208,009,016,141,017,208,088,096,000
2050 DATA 162,009,202,208,253,096,000,000,000,000,000,000,000,000,000,000
2060 DATA 173,017,208,041,239,141,017,208,169,000,141,013,221,169,255,141
2070 DATA 003,221,173,002,221,009,004,141,002,221,173,000,221,041,251,141
2080 DATA 000,221,169,000,133,251,169,096,133,252,160,000,120,177,251,141
2090 DATA 001,221,032,236,003,200,208,245,230,252,169,128,197,252,208,237
2100 DATA 173,017,208,009,016,141,017,208,088,096,000,000,000,000, 000,000
2110 DATA 162,012,202,208,253,096
```

Rajattoman laaja liitäntä mikrolta ulkomaailmaan

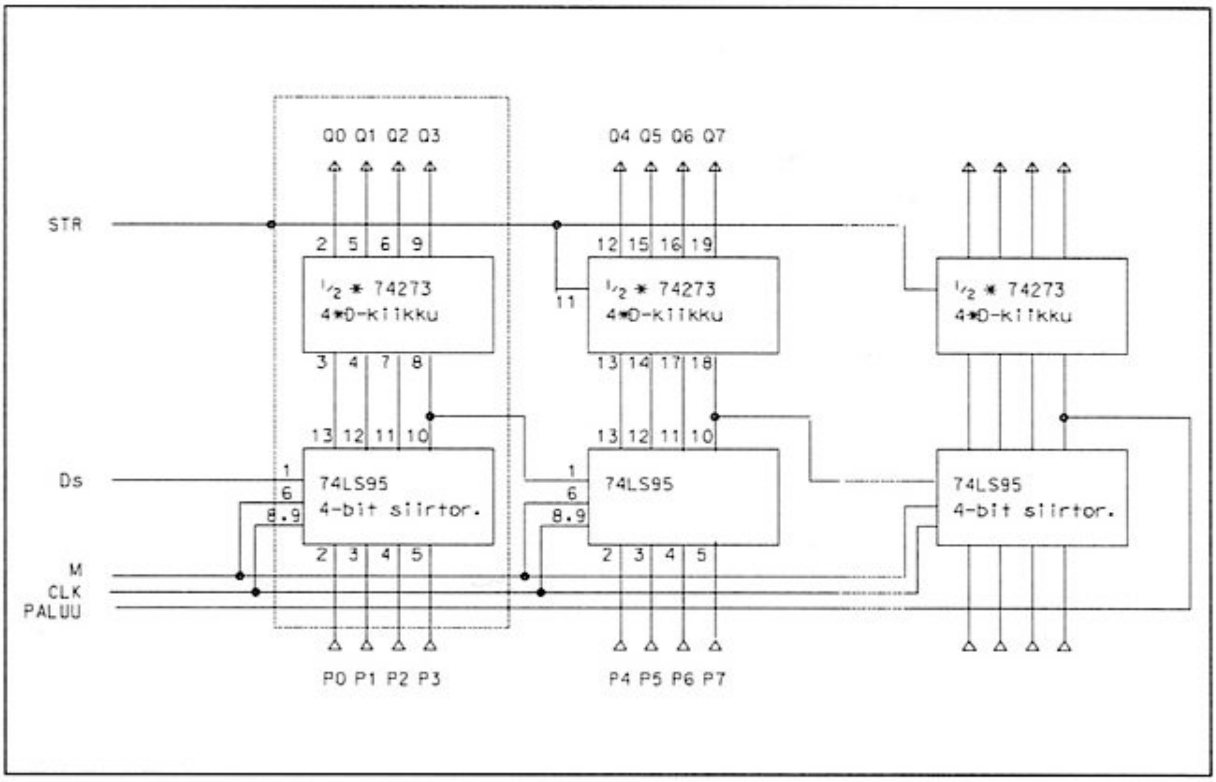
Usein mikroon omia lisälaitteita rakennettaessa tulee vastaan tilanne, jossa uutta laitetta ohjataan niin usealla binäärisignaalilla, etteivät mikron vakiovarusteisiin kuuluvat liittimet riitä. Pienissä kotimikroissa on yleensä käytettävissä korkeintaan pari kappaletta muutaman ohjausnastan sisältävää käyttäjänporttia. Esimerkiksi Commodore 64:n käyttäjällä on ilman lisälaitteita käytössään alle 20 vapaasti ohjelmoitavaa liitäntänastaa, joten releohjainten, ovisilmien ja muiden vempainten lukumäärän kasvaessa tulee väkisinkin eteen pulma: mistä lisää liittimiä.

Useimmiten mikrojen kyljistä löytyviä liitäntänastoja ja käytettäviä ohjelmia yhdistää liittimen tarkka osoite koneen muistiavaruudessa. Esimerkiksi C64:n käyttäjän portin nastoja C–L ohjataan ja luetaan muistiosoitteen 56577 sisältämien kahdeksan bitin kautta. Käyttäjänportin kaltaisia liittimiä on suhteellisen helppo rakentaa suoraan prosessoriväylään liittämällä.

Tällaisilla lisärakennelmilla on kuitenkin runsaasti varjopuolia. Laajennukset on kytkettävä suoraan prosessorin herkkiin osoite- ja dataväyliin, joihin virheellisen kytkennän suorittamalla voi helposti tuhota koneensa. Erilaiset laajennusmoduulit voivat myös käyttää portin kanssa samoja muistipaikkoja, mikä häiritsee sekä moduulin että oman kytkennän toimintaa ja pahimmassa tapauksessa rikkoo ne.

Tässä esitettävä kytkentä tarvitsee toimiakseen viisi kappaletta vapaasti ohjattavia liitäntänastoja. Vastalahjaksi se tarjoaa innokkaan rakentelijan käyttöön periaatteessa rajattoman pitkän liitinketjun. Kytkennän salaisuus on tavassa, jossa tietoa sen ja mikron välillä siirretään. Esimerkiksi 16 bittiä sisältävän portin tiedot siirretään yhtä johdinta pitkin bitti kerrallaan 16 bitin sarjana. Niinpä porttia laajennettaessa voi laajennukset kytkeä edellisten perään eikä mikron puolella tarvitse kuin kertoa ohjausohjelmalle portin uusi koko.

Tällaisen ratkaisun huonoin puoli on se, että portin koon kasvaessa sen toiminta hidastuu. Nopeudella tarkoitetaan tässä sitä, miten nopeasti portin loogisia jännitteitä saa vaihdettua tai miten nopeassa tahdissa jännitteitä tunnustelevien nastojen arvoja saa luettua. Hitaudesta ei kuitenkaan yleensä ole haittaa,



Kuva 1. Liitäntäpiirikortin kytkentäkaavio.

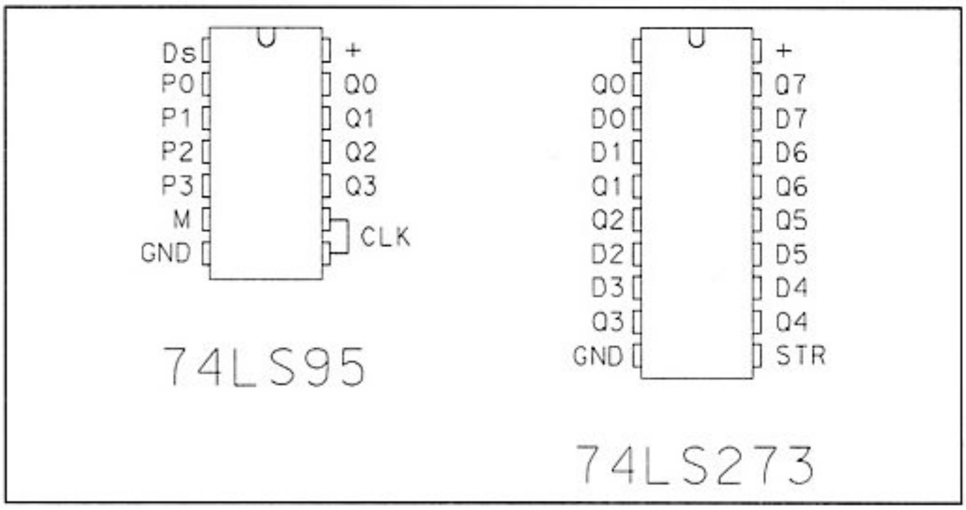
ellei porttiin aiota kytkeä digitoijia tai muita erittäin nopeita laitteita. Nopeutta voi lisätä huomattavasti kääntämällä oheisen Basic-ohjelman konekielelle.

Laite on esitetty tässä vanhan veteraanin VIC20:n joystick-porttiin sopivana, mutta sopivalla liittimellä ja esimerkkiohjelman osoitteita muuttamalla se sopii helposti myös muihin mikroihin, vaikkapa C64:n joystick-portiin tai mieluummin käyttäjän porttiin. Ainut vaatimus mikron puolelta on em. viiden vapaan nastan olemassaolo.

Kytkennän lohko- ja kytkentäkaavio on esitetty kuvassa 1. Sen toiminnan ymmärtäminen edellyttää muutamien digitaalipiirien toiminnan tuntemista. Koko laitteessa käytetään vain kahta piirityyppiä: D-kiikkua ja siirtorekisteriä. D-kiikun toiminta selvitettiin jo kirjan alkupuolella. Kytkennässä käytetään siirtorekisteriä nimeltään 74LS95 (kuva 2), joka toimii seuraavasti:

Jos otto M=1, niin piiri toimii kuin 4-bittinen D-kiikku. Kellosignaalin (CLK) putoaminen ykkösestä nollaan saa ottojen p0–p3 arvot siirtymään antoihin q0-q3. Tätä toimintaa kutsutaan siirtorekisterin rinnakkaislataukseksi, koska jokaiseen rekisterin bittiin siirretään tieto yhtäaikaa rinnakkaisia johtimia pitkin.

Jos taas M=0, niin samanlainen kellon muutos saa antojen arvot siirtymään yhdellä oikealle. Annon q2 arvo siis siirtyy q3:een, q1:n q2:een jne. Antoon q0 otetaan arvo erillisestä otosta Ds. Ottojen p0–p3 arvot eivät tällöin mitenkään



Kuva 2. Käytettävien mikropiirien nastajärjestykset.

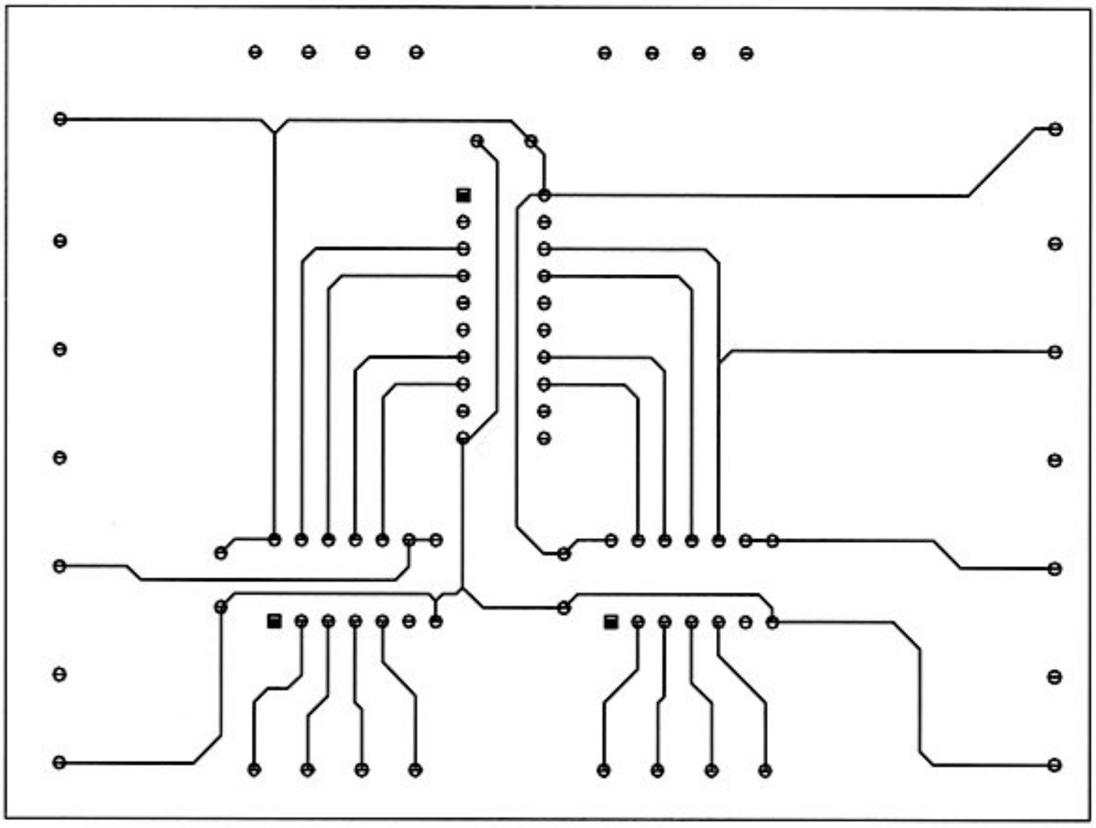
vaikuta piiriin. Siirtorekistereitä voidaan kytkeä mielin määrin sarjaan yhdistämällä aina edellisen piirin q3 seuraavan Ds:ään.

Rakennettava porttiketju muodostuu neljän bitin rakenneyksiköistä, joista kukin tarjoaa neljä D-kiikun antoa ohjaamaan muita laitteita. Kuvassa 1 on ensimmäinen kokonainen rakenneyksikkö ympäröity katkoviivalla. D-kiikut saavat tietonsa siirtorekistereiden annoilta, joihin ne ensin siirretään jonossa bitti kerrallaan edellä mainittua oikealle siirtoa käyttäen.

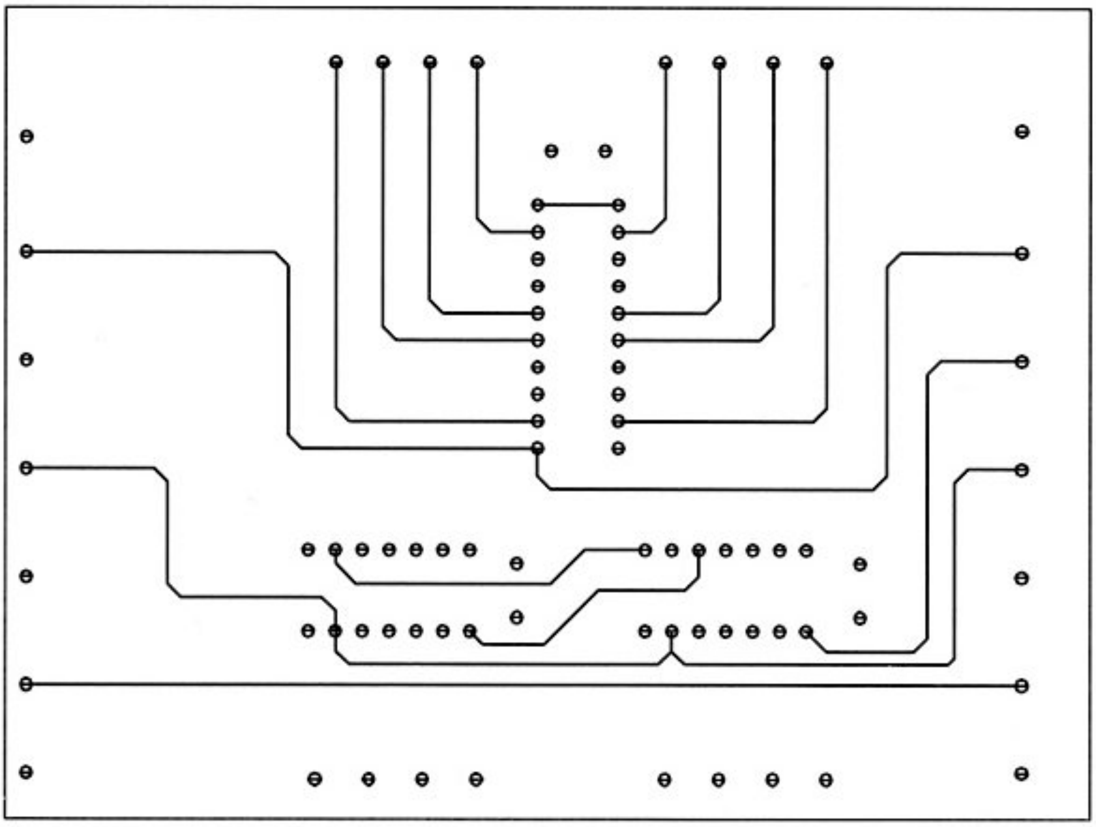
Lisäksi siirtorekistereiden rinnakkaislatausottoja käytetään loogisia tiloja tunnustelevina lukunastoina. Lukua varten tuodaan ketjun viimeisen siirtorekisterin q3:lta paluujohdin mikron luettavaksi. Kun nyt kaikki siirtorekisterit ladataan rinnakkaisotoistaan yhtä aikaa, voidaan kaikkien bittien arvot myöhemmin tutkia siirtämällä ne yksi kerrallaan oikealle kohti viimeistä q3:a ja tunnustelemalla samalla sen arvoa paluujohtimella.

Yksi rakenneyksikkö koostuu näin ollen neljästä D-kiikusta, ja neljän bitin siirtorekisteristä. Kuvien 3 ja 4 piirilevy sisältää kaksi rakenneyksikköä, jotka on rakennettu kahta 74LS95 siirtorekisteriä ja yhtä kahdeksan D-kiikkua sisältävää 74273:a (kuva 2) käyttäen. Huomaa, että D-kiikkujen kello-otto on sarjarekisterin kellon kanssa sekaannuksen välttämiseksi nimetty STR:ksi.

Kuten edellä jo mainittiin, tarvitsee koko ketju viisi ohjaavaa nastaa. Näistä yhdellä luetaan viimeiseltä siirtorekisteriltä tulevaa paluujohdinta ja toisella syötetään tietoa ensimmäisen rekisterin ottoon Ds. D-kiikkujen yhtäaikaiseen liipaisuun tarvitaan yksi johdin. Loput kaksi tarvitaan siirtorekistereiden liipaisuun ja toimintamoodin (otto M) asetukseen.



Kuva 3. Liitäntäpiirikortin komponenttipuoli.



Kuva 4. Liitäntäpiirikortin juotospuoli.

Ohjausohjelman runko

Listauksessa 1 esitetty ohjelmarunko sisältää kolme aliohjelmaa, joilla mielivaltaisen kokoista ketjua voi ohjata. Rivillä 0 määritellään tiedon siirrossa tarvittavat kaksi taulukkoa, ja rivillä 1 ketjussa olevien laiteyksiköiden lukumäärä. Käytettävien viiden nastan osoitteet, bittinumerot ja suuntarekisteriosoitteet on määritelty riveillä 2–6. Mikäli ketju kytketään muualle kuin VIC20:n joystickporttiin, tulee nämä arvot muuttaa oikeiksi. Tarvittavat tiedot löytyvät esimerkiksi C64:lle Programmer's Reference Guidesta (muistikartta), kunhan on vain saanut päätettyä, mitä nastoja ja portteja käyttää.

Riveillä 1 000–1 030 asetetaan valittujen nastojen suunnat oikeiksi (paluujohdinta luetaan, muita kirjoitetaan). Tätä aliohjelmaa tulee kutsua mieluiten heti ohjelman alussa, kuten rivillä 10 tehdään.

D-kiikkujen asetusaliohjelma on riveillä 2 000–2 110. Ennen sen kutsumista tulee taulukkoon D() asettaa kullekin 4 bitin yksikölle menevä tieto (luku väliltä 0–15, 0=kaikki kiikut alas, 15=kaikki kiikut ylös). D(1):n tulee sisältää ketjun ensimmäisen yksikön tieto, D(2):n toisen jne. Käskyllä GOSUB 2000 taulukkoon asetettu tieto sitten siirtyy kiikkujen antoihin.

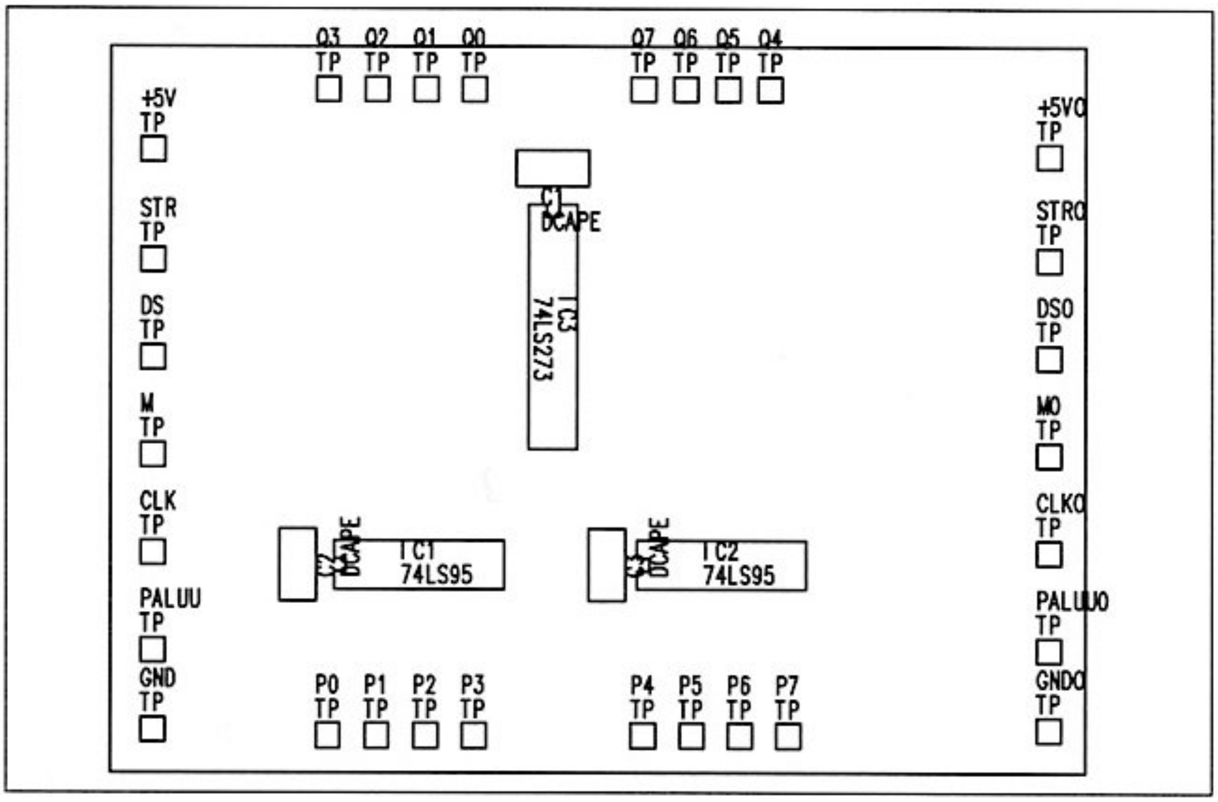
Käskyn GOSUB 3000 jälkeen voi taulukosta I() lukea kunkin siirtorekisterin rinnakkaisottojen senhetkiset arvot. Taulukon luvut noudattavat samaa sääntöä kuin taulukossa D(), eli I(1) sisältää ensimmäisen yksikön tiedot (0..15, 0=kaikki otot alhaalla, 15=kaikki otot ylhäällä).

Ketjun pidennys

Ketjun pidennys onnistuu helposti rakentamalla lisää kuvan 3 mukaisia levyjä, ja kytkemällä ne lohkokaavion 1 mukaisesti sarjaan. Viimeisen yksikön q3 tulee kytkeä paluujohtimeen, jotta lukunastojen arvot saadaan mikrolle. Ketjun käyttöjännite voidaan ottaa mikrolta, jos ketjun pituus ei kasva kovin suureksi (alle 5 yksikköä). Tavallisesta ns. Atari-tyyppisestä joystick-liittimestä käyttöjännitteen saa nastoista 7 (+5V) ja 8 (maa). Käyttäjän portilla vastaavasti +5 V löytyy nastalta 2 ja maa nastalta 1.

Mikäli erilliset yksiköt sijoitetaan kovin kauas toisistaan (yli metri), tulisi toimintahäiriöiden välttämiseksi yksiköiden yhdistämiseen käyttää metalliverkolla suojattua kaapelia.

Laite voidaan rakentaa esimerkiksi kuvien 3 (komponenttipuolen johdotus), 4 (kääntöpuolen johdotus) ja 5 (osasijoittelu) piirilevylle.



Kuva 5. Liitäntäpiirikortin osasijoittelu.

Komponenttiluettelo

IC1,IC2	74LS95
IC3	74LS273
C1	100 nF

LISTAUS 1

```
0 DIM D(10),I(10) : REM MUUTA DIMENSIOITA JOS YKSIKÖITÄ  10
1 NR = 2 : REM LAITEYKSIKOIDEN LUKUMAARA
2 KM = 37137 : KB = 3 : KS = 37139 : REM SIIRTOREK. KELLOSIGNAALI
3 MM = 37137 : MB = 5 : MS = 37139 : REM MOODI-SIGNAALI
4 SM = 37137 : SB = 2 : SS = 37139 : REM SYOTTOSIGNAALI
5 OM = 37152 : OB = 7 : OS = 37154 : REM PALUUSIGNAALI
6 FM = 37137 : FB = 4 : FS = 37139 : REM D-KIIKKUJEN LIIPAISUSIGNAALI
10 GOSUB 1000 : REM SUUNTIEN ASETUS
** Tähän väliin pääohjelma **
1000 REM ** SUUNTA **
1010 POKE SS,(PEEK(SS) OR (2^SB))
1011 POKE KS,(PEEK(KS) OR (2^KB))
```

```
1012 POKE MS,(PEEK(MS) OR (2^MB))
1013 POKE FS,(PEEK(FS) OR (2^FB))
1014 POKE OS,(PEEK(OS) AND (255-2^OB))
1020 POKE MM,(PEEK(MM) AND (255-2^MB))
1030 RETURN
2000 REM ** ULOS **
2010 FOR I = 1 TO NR   : REM KAY LAPI KAIKKI LAITEYKSIKOT
2020 FOR J = 0 TO 3    : REM KAY LAPI KAIKKI 4 BITTIA
2030 DB = (2^SB)*(D(I) AND 2^J)/(2^J) : REM MUODOSTA TAULUKOSTA 1 BITTI
2040 POKE SM,(PEEK(SM) AND (255-(2^SB))) : REM ASETA SYOTETTAVA BITTI
2050 POKE SM,(PEEK(SM) OR DB)        : REM TYRKYLLE
2060 POKE KM,(PEEK(KM) OR (2^KB)) : REM LIIPAISE SARJAREKISTEREIDEN
2070 POKE KM,(PEEK(KM) AND (255-2^KB)): REM BITIT YHDELLA ETEENPAIN
2080 NEXT J,I
2090 POKE FM,(PEEK(FM) AND (255-2^FB))   : REM LIIPAISE D-KIIKUILLE
2100 POKE FM,(PEEK(FM) OR (2^FB))     : REM SARJAREKISTEREIDEN ARVOT
2110 RETURN
3000 REM ** SISAAN **
3010 POKE MM,(PEEK(MM) OR (2^MB))    : REM SARJAREKISTERIT RINNAKKAISMOODIIN
3020 POKE KM,(PEEK(KM) OR (2^KB))   : REM LIIPAISE SARJAREKISTERIT TAYTEEN
3030 POKE KM,(PEEK(KM) AND (255-2^KB))
3040 POKE MM,(PEEK(MM) AND (255-2^MB)) : REM SARJAREKISTERIT SARJAMOODIIN
3050 FOR I = NR TO 1 STEP -1    : REM KAY LAPI KAIKKI LAITEYKSIKOT
3060 I(I) = 0           : REM YKSIKON ARVO
3065 FOR J = 3 TO 0 STEP -1  : REM 4 BITTIA
3070 DB = (PEEK(OM) AND (2^OB))/(2^OB) : REM LUE YKSI BITTI
3080 I(I) = I(I)+DB*(2^J)   : REM LISAA PAINOTETTU BITTI YKSIKKOON
3090 POKE KM,(PEEK(KM) OR (2^KB)) : REM LIIPAISE SARJAREKISTERIEN
3100 POKE KM,(PEEK(KM) AND (255-2^KB)) : REM BITTEJA YHDELLA ETEENPAIN
3110 NEXT : NEXT
3120 RETURN
```

Yleiskäyttöinen digitaalilaskuri

Tämä kytkentä on monikäyttöinen digitaalinen laskuri, jota voi käyttää osana muita käytännön laitteita tai soveltaa sellaisenaan. Laskurikytkentä laskee sille tuotavien digitaalipulssien määrää. Laskettujen pulssien määrä muutetaan binääriluvuksi, jonka voi lukea laskimelta esimerkiksi mikroon jatkokäyttöä varten. Erillisellä signaalilla laskurin voi nollata tai asettaa johonkin muuhun ennalta määrättyyn alkuarvoon.

Laskureilla on monia käyttökohteita. Yksinkertaisimmillaan sitä voi soveltaa laskemaan esimerkiksi oveen rakennetun kulunvalvontakytkimen sulkeutumisien lukumäärää. Erilaisissa digitaalimittauslaitteissa XTlaskuri on usein oleellinen osa. Esimerkiksi signaalin taajuus voidaan mitata nollaamalla laskuri ja antamalla sitten laskurin laskea signaalin pulssien määrää tasan yhden sekunnin ajan.

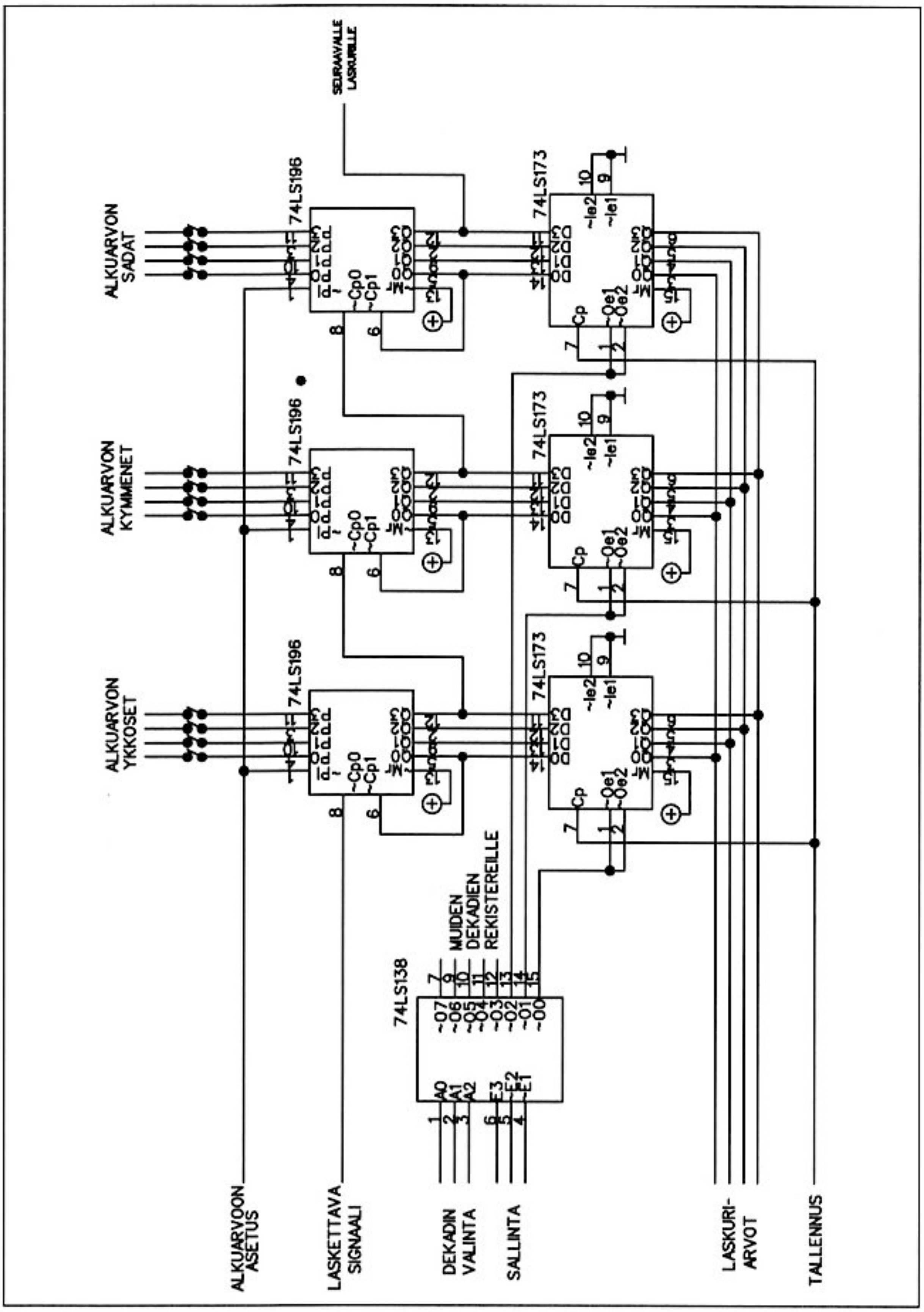
Laskuria voidaan käyttää myös sekunttikellona mittaamalla tarkasti yhden hertsin signaali ja käyttämällä laskurin laskemaa binäärilukua suoraan sekunttilukemana. Kilohertsin signaalia käyttämällä saadaan kulunut aika sekunnin tuhannesosina jne.

Nykyisin hyvinkin monimutkaisia laskureita saa suoraan yhtenä mikropiirinä. Tämä laskuri on kuitenkin toteutettu tavallisilla LS-TTL-piireillä, millä on saatu aikaan mukavia muokkausmahdollisuuksia.

74LS196

Laskettava TTL-signaali tuodaan kuvassa 1 olevan kytkentäkaavion vasemmanpuolimmaiselle 74LS196-piirin nastalle 8. 74LS196 sisältää kaksi kappaletta erillisiä laskureita, joista toinen laskee kahteen ja toinen viiteen. Nämä laskurit voidaan kytkeä sarjaan, jolloin tuloksena on kymmeneen asti laskeva piiri. Laskuri kasvattaa rekisterissä olevaa arvoaan aina kellosignaalin laskevalla reunalla.

Laskettu määrä saadaan suoraan 4-bittisenä binäärilukuna nastoista 5, 9, 2 ja 12. 74LS196 voidaan asettaa haluttuun arvoon viemällä asetettava binääriluku



Kuva 1. Laskuripiirin kytkentäkaavio.

Arvo	Q3	Q2	Q1	Q0	Huomautus
0	0	0	0	0	Laskurin nollakohta
1	0	0	0	1	
2	0	0	1	0	
3	0	0	1	1	
4	0	1	0	0	
5	0	1	0	1	
6	0	1	1	0	
7	0	1	1	1	
8	1	0	0	0	Q3:n nouseva reuna
9	1	0	0	1	Laskuri pyörähtää ympäri, seuraava arvo taas 0
0	0	0	0	0	Q3:n laskeva reuna

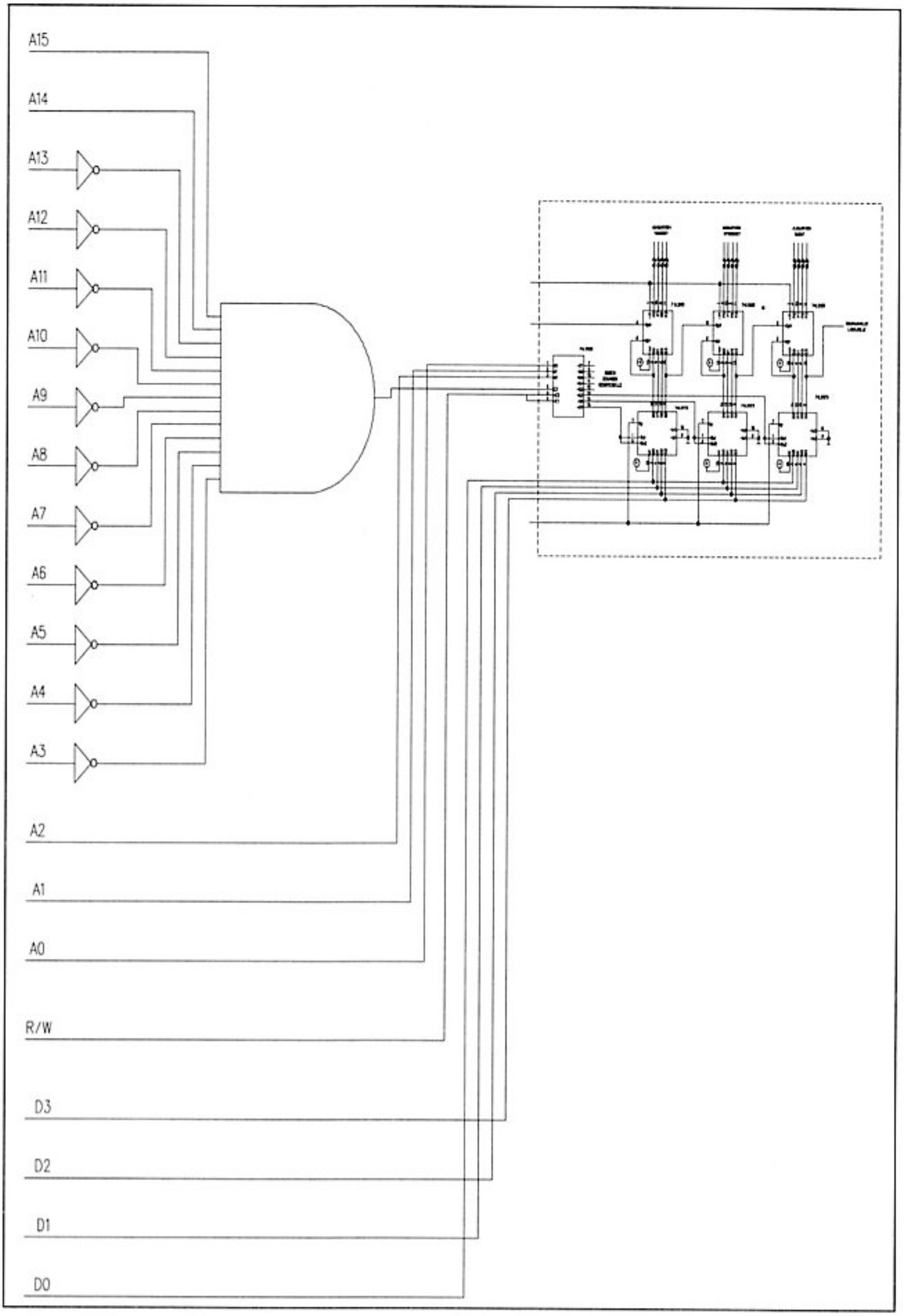
Taulukko 1.

nastoille 4, 10, 3 ja 11 ja antamalla sen jälkeen ns. rinnakkaislatauspulssi nastaan 1. Kun laskuri pääsee yhdeksään asti, pyörähtää se seuraavalla laskentasignaalilla ympäri ja uudeksi arvoksi tulee nolla.

Jotta laskuri saadaan ymmärtämään suurempia lukumääriä 74LS196-kymmenlaskureita kytketään useita peräkkäin. Tämä tapahtuu viemällä edellisen laskurin eniten merkitsevä bitti (Q3) aina seuraavan laskurin kellosignaaliksi (Cp0). Tällainen menettely toimii, sillä jos tarkastellaan kymmenlaskurin antoja yhden kokonaisen kierroksen aikana, huomataan, että Q3 putoaa sopivasti ykkösestä nollaan (laskuri kasvattaa arvoaan aina laskevalla reunalla) kierroksen aikana vain juuri ympäripyörähdyksen aikana taulukon 1 mukaisesti.

Itse asiassa 74LS196:n tyyppistä piiriä voidaan siis käyttää myös signaalitaajuuden kymmenjakajana. Jos mitattavan signaalin taajuus on esimerkiksi 1 MHz (miljoona laskevaa reunaa sekunnissa), Q3:n reuna laskee tasan 100000 kertaa sekunnissa.

Yhdellä lisäpiirillä laskurin kapasiteetti aina kymmenkertaistuu. Kahdella laskurilla saadaan siis laskettua sataan asti ja kuvan 1 mukaisella kolmen laskuripiirin laitteella aina tuhanteen asti. Kymmenlaskureita voi tarvittaessa kytkeä lisää SEURAAVALLE LASKURILLE -tekstillä merkittyyn pisteeseen.



Kuva 2. Laskuripiirin kytkeminen mikrotietokoneen väylään.

74LS273

Laskureiden tulosnastoihin Q0–Q3 on kytketty D-kiikut 74LS273, joihin laskureiden tulokset voidaan tallentaa. Välitallennusta tarvitaan esimerkiksi jos laskuria käytetään taajuusmittarina, koska lopullinen mittaustulos saadaan laskureilta vasta mittausjakson lopussa. Laskureiden tulokset tallennetaan D-kiikkuihin tuomalla pulssinreuna nastaan 7 (signaali TALLENNUS). 74LS273-kiikut sisältävät lisäksi 3-tilapuskurit, joten kaikkien kiikkujen annot voi kytkeä yhteen väyläksi ja ohjata ns. output enable -signaaleilla vain yhden kiikkusarjan annot aktiivisiksi.

Tämä tapahtuu yksi-kahdeksasta-valitsimelle 74LS138. 74LS138:lle tuodaan ohjaussignaaleina kolmen bitin binääriluku (A0–A2), joka valitsee yhden annoista (Q0–Q7) menemään nollatilaan muiden pysyessä ykkösinä. Lisäksi sallintasignaaleilla (E1–E3) voidaan valita, onko annoista ylipäätään mikään nollatilassa eli onko piiri "päällä" vai ei.

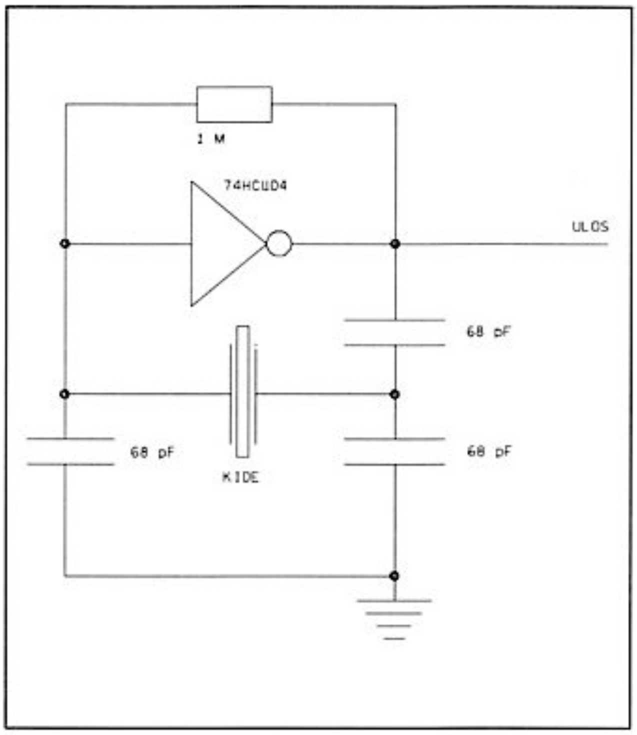
Tämä järjestely mahdollistaa koneen helpon liittämisen mikrotietokoneen väylään (kuva 2). DEKADIN VALINTA -signaalit kytketään suoraan koneen osoiteväylän alimpiin bitteihin. Näin saadaan aikaan laskurin peräkkäisiä dekadeja vastaavien rekistereiden sijoittuminen tietokoneen peräkkäisiin muistipaikkoihin.

LASKURIARVOT-väylä kytketään suoraan koneen dataväylän neljään alimpaan bittiin. Laskuria lukevassa ohjelmassa tulee muistaa poistaa ylempien bittien vaikutus lukuarvoihin esimerkiksi pudottamalla ylimmät bitit pois JA-operaattorilla heti luvun jälkeen.

SALLINTA-signaaleille tuodaan sopivalla porttikokoonpanolla tietokoneen ylemmistä osoitesignaaleista muodostetut signaalit, siten että E3=1, E2=E1=0 silloin ja vain silloin, kun prosessori lukee sitä muistialuetta johon laskurin halutaan mikron muistiavaruudessa sijoittuvan. Kuvan 2 esimerkissä laite on sijoitettu muistiavaruuden osoitteisiin C000–C007.

Jos laskuria ei kytketä mikroon, vaan tuloksia käytetään sellaisenaan, voi LASKURIARVOT -väylän ja valitsimen 74LS138 jättää pois ja kytkeä 74LS273-piirien sallintasignaalit (nastat 1 ja 2) pysyvästi maahan.

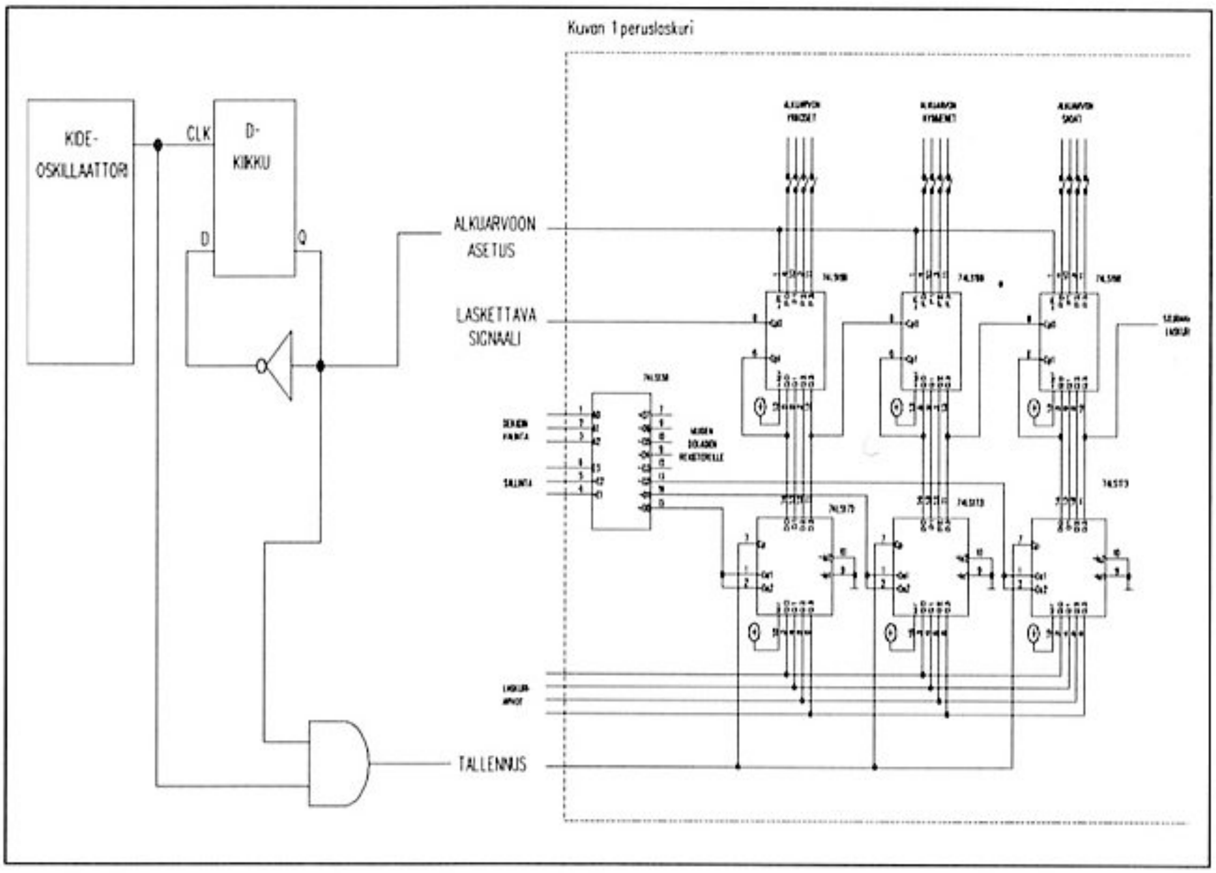
Taajuuslaskuri- ja kellotoiminnoissa tarvittava tarkka ns. vertailusignaali muodostetaan yleensä kvartsikideoskillaattorilla.XT Kvartsikiteillä on erittäin tarkasti vakaana pysyvä ns. ominaisvärähtelytaajuus, jota käytetään hyväksi taajuudeltaan tarkoissa oskillaattoreissa. Kideoskillaattori voidaan rakentaa esimerkiksi kuvan 3 kytkennällä.



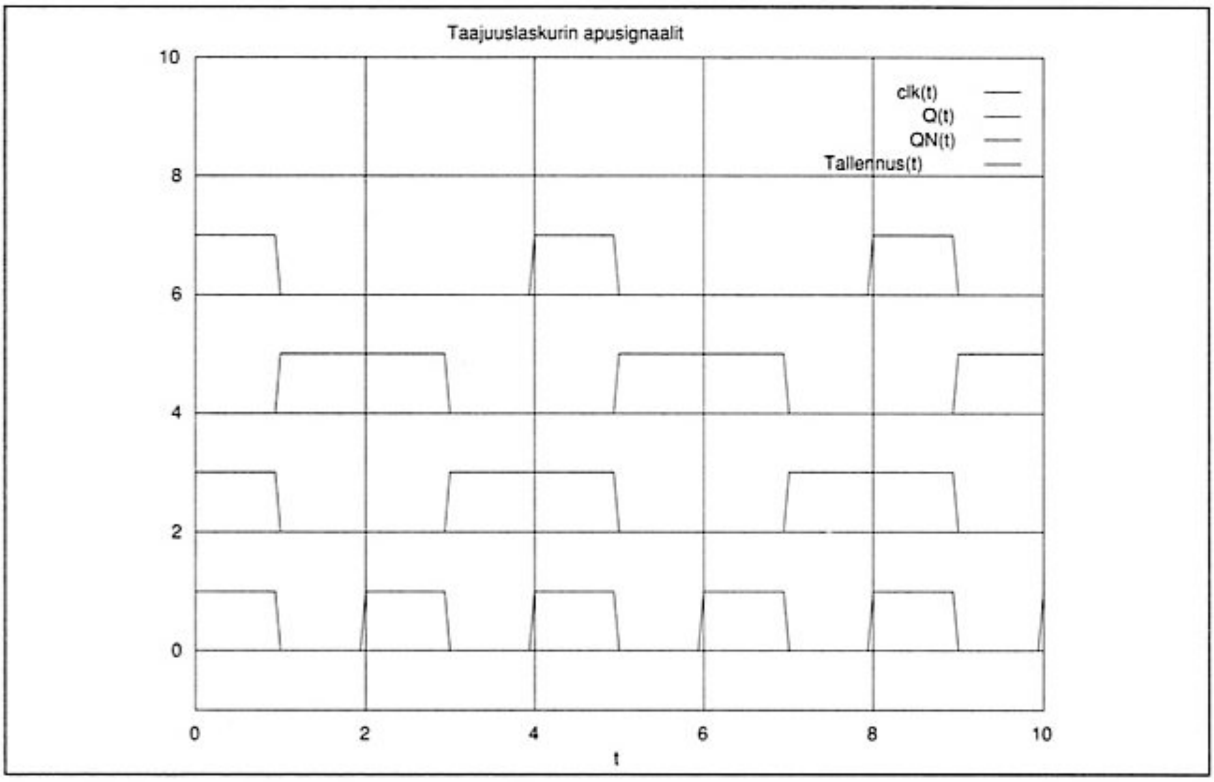
Kuva 3. Kideoskillaattori.

Kideoskillaattoreista saatavat taajuudet ovat yleensä liian suuria sellaisenaan käytettäviksi, koska kiteiden ominaistaajuudet ovat varsin korkeita (vähintään satoja kilohertsejä). Taajuutta voi näissä tapauksissa pienentää käyttämällä esimerkiksi edellä mainitulla tavalla 74LS196-piiriä kymmenjakajana.

Taajuusmittariksi laskuri kytketään kuvan 4 mukaisesti. Mitattava taajuus tuodaan normaalisti LASKETTAVA SIGNAALI -nastaan. Kideoskillaattorilla kehitetään mittausjakson pituutta vastaava taajuus puolitettuna. D-kiikkua käyttäen oskillaattorin taajuus jaetaan puoleen ja nämä molemmat yhdistetään loo-



Kuva 4. Laskuripiirin käyttö taajuuslaskimena.



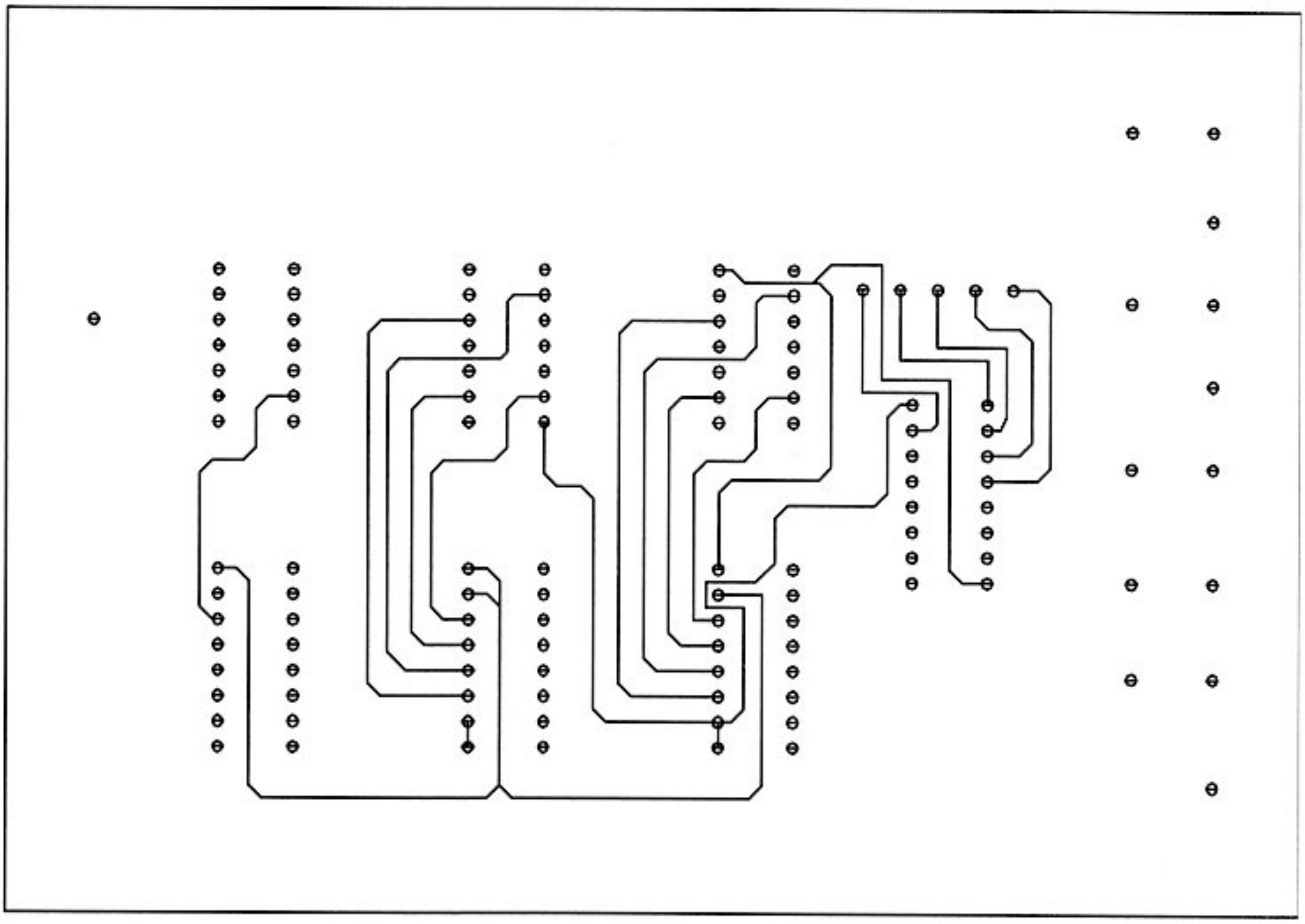
Kuva 5. Taajuuslaskurin apusignaalit.

gisella JA-portilla TALLENNUS-signaaliksi.

Taajuuslaskurin ajoituskaavio on esitetty kuvassa 5. Hetkellä 1 signaali Q (=ALKUARVON ASETUS) putoaa nollaan, jolloin alkuarvot siirtyvät rekistereihin. Pakkoasetus pitää alkuarvon rekistereissä aina hetkeen 3 asti, jolloin Q nousee ylös. Tällöin alkavat laskurit laskea mitattavan signaalin tahdissa ylöspäin aina hetkeen 4 asti, jolloin TALLENNUS-signaali nousee ykköseksi. Tämä aiheuttaa laskureiden senhetkisen arvon siirtymisen D-kiikkuihin 74LS273-piireihin. Tämän jälkeen hetkellä 5 signaali Q putoaa jälleen alas, jolloin ollaan hetkeä 1 vastaavassa tilanteessa.

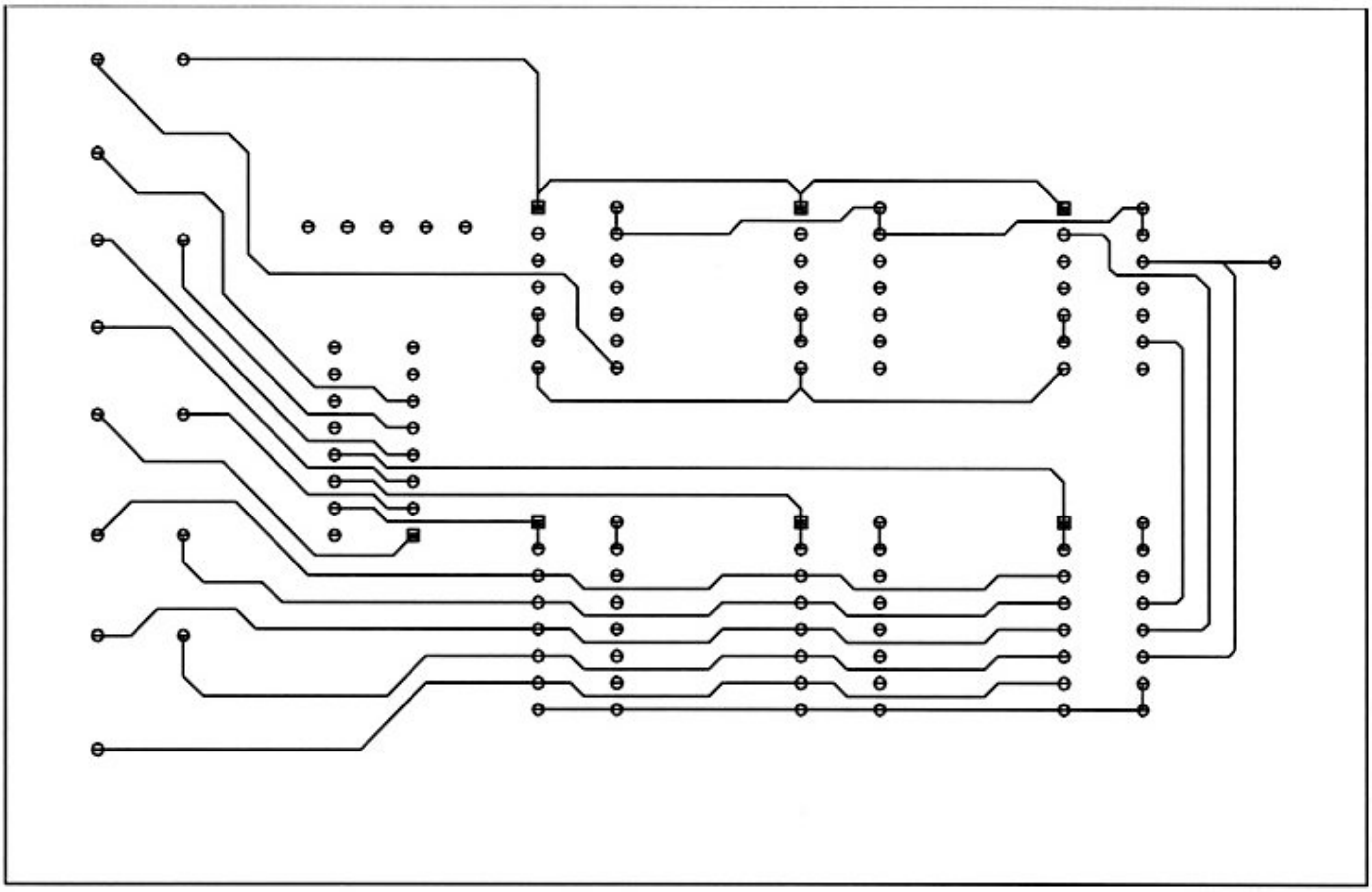
Koska D-kiikkujen arvo pysyy muuttumattomana aina seuraavaan lataukseen asti, voidaan niistä koska tahansa lukea edellisen kierroksen aikana hetkien 3 ja 4 välissä havaittujen laskevien reunojen määrä. Jos nyt kideoskillaattorilta saatava taajuus säädetään esimerkiksi puoleen hertsiin, on hetkien 3 ja 4 väliin jäävä aika tasan 1 sekunti. Tällöin D-kiikuista saatava tulos on mitattavan taajuuden arvo hertseinä.

Jos mitattava taajuus on suuri, voidaan mittausaikaa lyhentää kasvattamalla kideoskillaattorin taajuutta. Esimerkiksi oskillaattoritaajuudella XT500 kHz tulisi yhden mittausvälin pituudeksi 1 µs, jolloin laskurilta saatu taajuuslukema on miljoonasosa todellisesta taajuudesta, toisin sanoen saatu tulos on mitattu taajuus megahertseinä ilmaistuna.

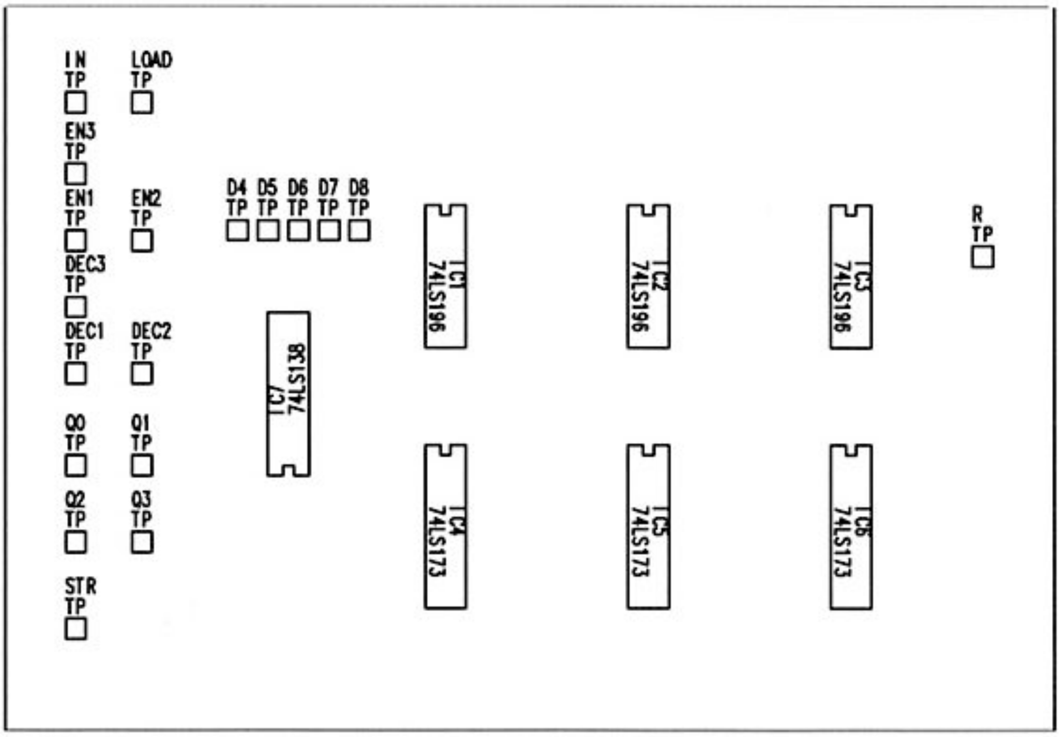


Kuva 6. Laskuripiirikortin komponenttipuoli.

Kuvan 1 kytkentä voidaan rakentaa esimerkiksi kuvissa 6 (komponenttipuolen johdotus), 7 (kääntöpuolen johdotus) ja 8 (osasijoittelu) esitetylle piirilevylle.



Kuva 7. Laskuripiirikortin juotospuoli.



Kuva 8. Laskuripiirikortin osasijoittelu.

Sarjaliikenneohjain

Monet nyt jo hieman vanhahtavat kotimikrot kärsivät eräästä puutteesta, joka aiheuttaa varsinkin modeeminkäyttäjille lisämenoja. Tämä ongelma on ns. sarjaliitäntä tai pikemminkin sen puute. RS-232-nimistä sarjaliitäntää käytetään yleisesti mikrojen ja sen oheislaitteiden kytkemiseksi toisiinsa.

Sarjaliitännän erikonen piirre bittinikkarin kannalta on siinä käytettävät jännitetasot. Digitaalielektroniikassa tullaan nykyisin useimmiten toimeen vain yhdellä viiden voltin käyttöjännitteellä. RS-232:ssa kuitenkin tarvitaan tietojen siirtämiseen erikoisia +3 – +12 ja -3 – –12 voltin jännitteitä.

Normaalisti tietokoneen piireissä loogista nollaa vastaa nollan voltin ja loogista ykköstä viiden voltin jännite. RS-232:ssa kuitenkin mikä tahansa jännite 3 ja 12 voltin väliltä tarkoittaa loogista nollaa ja mikä tahansa jännite –3 ja –12 voltin väliltä loogista ykköstä.

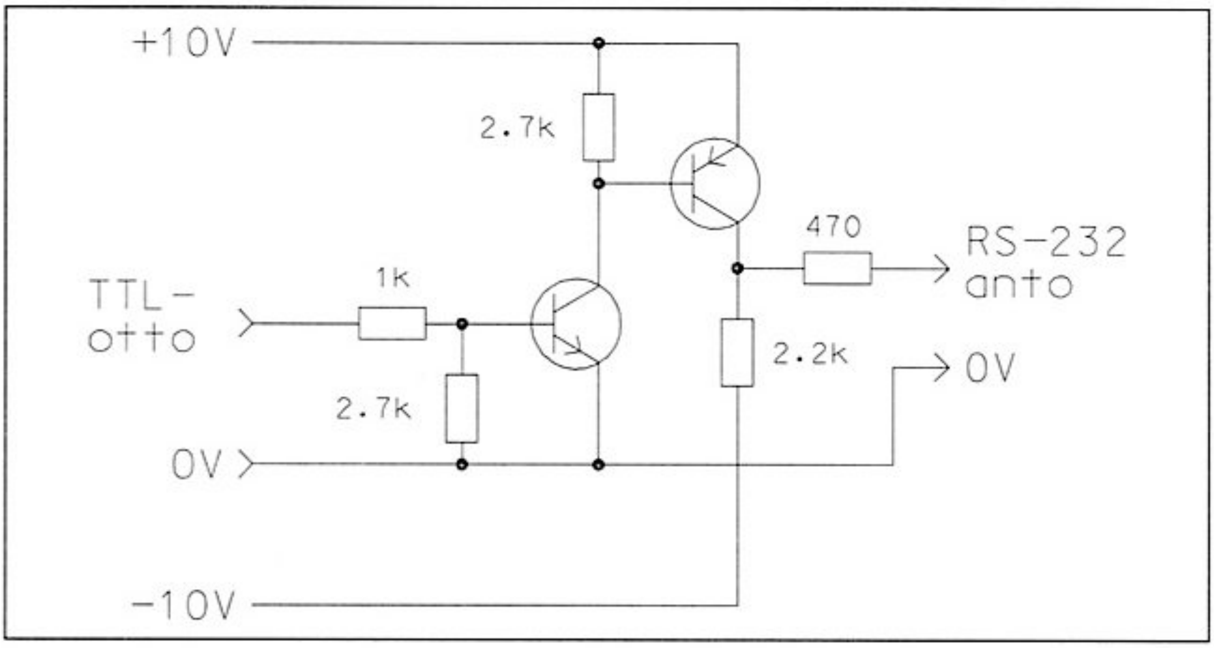
Tässä onkin syy siihen, miksi RS-232-liitäntä puuttuu joistakin mikroista. Kun muuten koneessa tullaan toimeen yhdellä viiden voltin jännitteellä, sarjaliitännän lisääminen olisi aikoinaan maksanut roimasti lisää, kun teholähteeseen olisi pitänyt lisätä negatiivisen jännitteen tuottava kytkentä.

Esimerkiksi Commodoren C64 ja VIC-20 ovat hyviä esimerkkejä tällaisista koneista. Commodoren suunnittelijat eivät kuitenkaan jättäneet RS-232:ta kokonaan pois. Itse asiassa koneet sisältävät täysin toimivan sarjaliitännän ohjelmistoineen, mutta niiden jännitteet on vain jätetty RS-232-standarinmukaisten sijasta TTL-tasolle.

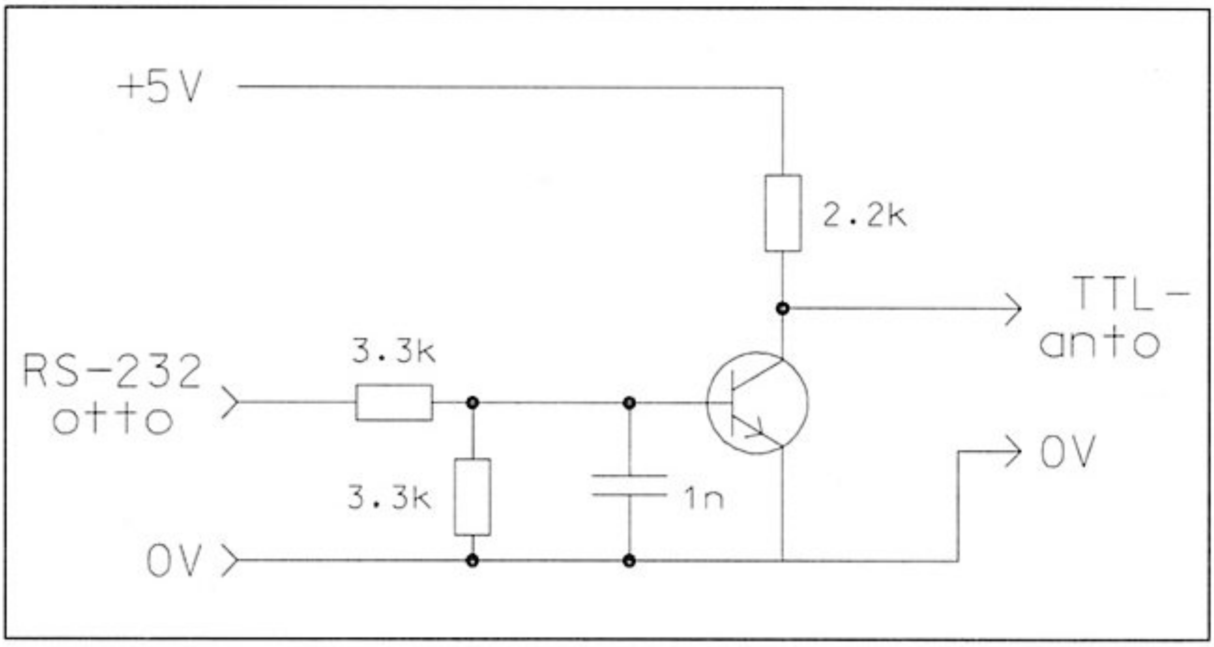
C64:ään ja VIC-20:een voikin liittää minkä tahansa RS-232-liitäntäisen laitteen sen jälkeen kun TTL-tasoiset jännitteet on muunnettu RS-232-tasolle.

Mikäli käytössä on jo esimerkiksi muista lisälaitteista saatavana symmetrinen jännitepari, esimerkiksi –10 ja +10 volttia, ei jännitemuuntimeen tarvita kuin muutama transistori, vastus ja kondensaattori. Kuvissa 1 ja 2 on esitetty yksinkertaiset muuntimet molempiin suuntiin. Kuvan 1 kytkentä muuntaa TTL-tasoisen signaalin RS-232-tasolle ja kuvan 2 kytkentä päinvastoin.

Nykyisin on saatavilla myös erittäin näppäriä mikropiirejä, joilla loogisten jännitetasojen muuntelu sujuu erittäin mukavasti. Eräs tällainen piiri on MAX232, joka sisältää kaksi kappaletta kumpaakin muunnintyyppiä. Piirin paras ominaisuus on kuitenkin sen sisältämä jännitteennostopiiri, joka tekee ulkoisen symmetrisen jännitelähteen tarpeettomaksi.

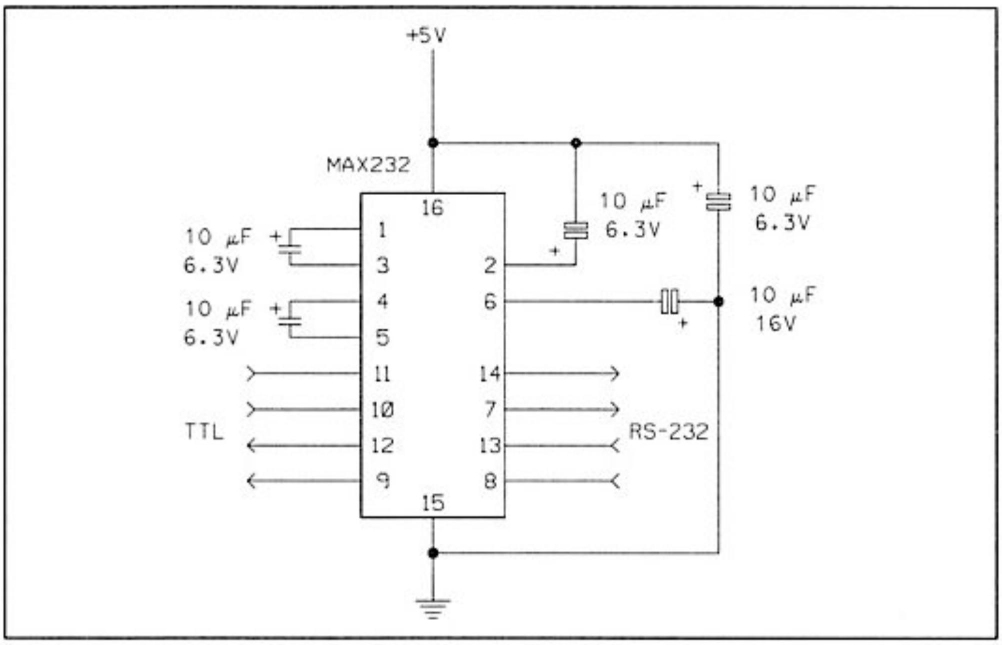


Kuva 1. TTL-RS-232-jännitemuunnin.



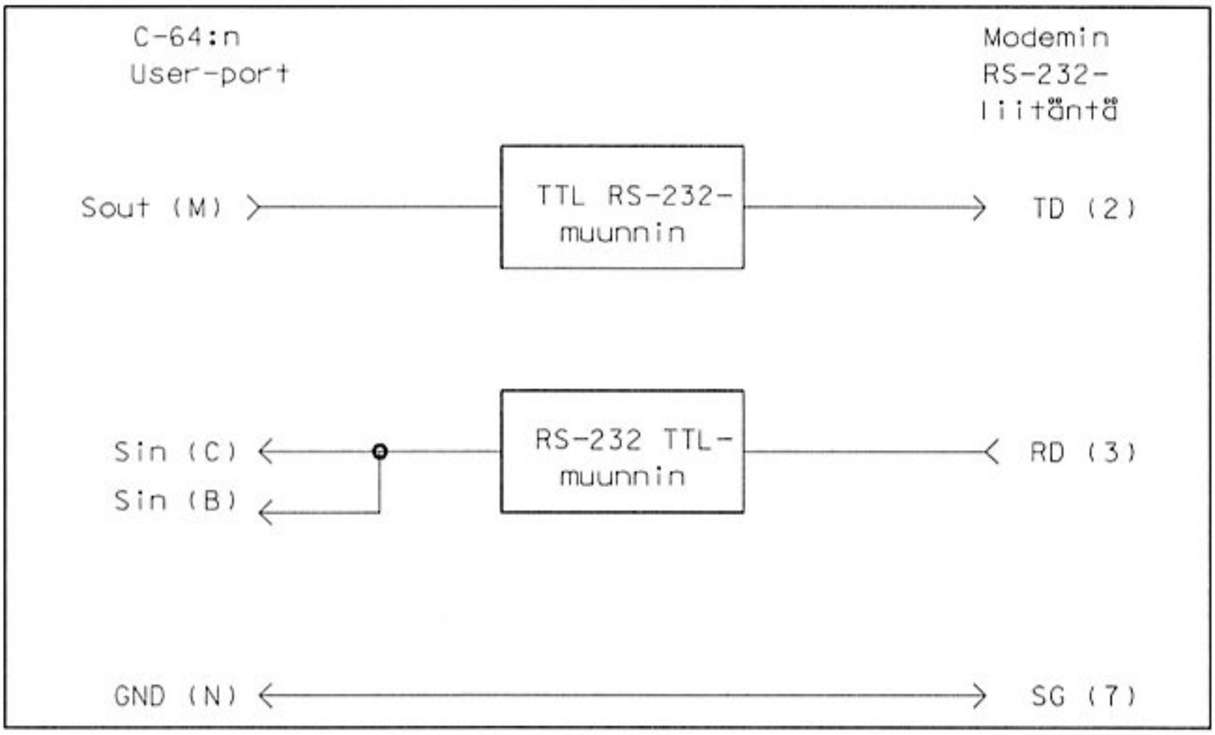
Kuva 2. RS-232-TTL-jännitemuunnin.

MAX232:lla voidaankin kuvan 3 mukaisesti toteuttaa neljän signaalin muunnokset vain normaalia viiden voltin jännitettä käyttämällä. Piirin nastoihin 11 ja 10 kytketään muunnettavat kaksi TTL-signaalia, jotka saadaan RS-232-tasoisina ulos nastoista 14 ja 7. Vastaavasti RS-232-signaalit syötetään nastoihin 13 ja 8, ja otetaan TTL-tasolle muunnettuina ulos nastoista 12 ja 9.



Kuva 3. MAX232-piirillä toteutettu kaksisuuntainen jännitemuunnin.

RS-232-liitäntä sisältää lukuisia signaaleja, joista vain kaksi siirtävät varsinaista tietoa. Muiden tehtävänä on hoitaa kytkettyjen laitteiden välinen ns. kättely . Kätellessään laitteet muun muassa pyytävät toiselta lupaa lähettää ja kertovat toisilleen milloin ne ovat valmiita ottamaan vastaan.



Kuva 4. Commodore 64:n ja modeemin välinen 3-lankainen RS-232-yhteys.

Periaatteessa jokaista sarjaliitännän signaalia varten tarvitaan yksi kuvan 1 tai 2 mukainen muunnin, mutta käytännössä harvat laitteet vaativat paria kolmea signaalia enempää. Usein riittää jopa ainoastaan varsinaista dataa siirtävien johtimien ja maajohtimen kytkeminen.

Tätä ns. 3-lankayhteyttä voidaan käyttää esimerkiksi C64:n ja modeemin välillä. Tällöin rakennetaan vain kaksi jännitemuunninta ja ne kytketään kuvan 4 mukaisesti. Kuvassa C64:n käyttäjän portin (user port) puolella johtimien nimet viittaavat Commodoren itsensä antamiin nimiin ja suluissa on annettu kyseisen käyttäjän portin nastan tunnus. Vastaavasti modeemin puolella nimet viittaavat RS-232-standardin signaalinimiin ja suluissa on annettu nastan numero silloin kun käytetään 25-napaista D-liitintä, joka on yleisin modeemeista löytyvä liitintyyppi.

Modeemista riippuen voidaan joskus joutua myös "huijaamaan" modeemia kytkemällä joitakin kättelysignaaleista pysyvästi jompaan kumpaan loogiseen tilaan. Tällainen virittely on kuitenkin jo aivan oma taiteenlajinsa.

Sanasto

4000-sarja

Edullinen ja yleinen CMOS-piiritekniikkaan perustuva digitaalimikropiirisarja.

6502

8-bittinen mikroprosessori, jota käytettiin yleisesti 80-luvun kotimikroissa. Esimerkiksi Commodoren VIC-20 ja Apple II sisältävät 6502:n.

74-sarja

Edullinen ja yleinen TTL-piiritekniikkaan perustuva digitaalimikropiirisarja. Alkuperäisestä 74-sarjasta tehdään myös tehonkulutukseltaan ja nopeudeltaan poikkeavia 74L-, 74LS-, 74S-, 74ALS- ja 74AS-sarjoja.

ALU

Aritmeettilooginen yksikkö (engl. Arithmetic Logic Unit). Tietokoneen keskusyksikön laskutoimituksia ja loogisia operaatioita suorittava yksikkö.

Ampeeri (A)

Sähkövirran voimakkuuuden yksikkö. Määritelty sähkövirraksi, joka kahdessa äärettömän pitkissä ja ohuissa yhdensuuntaisissa tyhjiössä yhden metrin päässä toisistaan olevissa johtimissa aiheuttaa niiden välille 2x10^-7 Newtonin voiman pituusmetriä kohti.

Analoginen

Käytetään kuvaamaan sähköistä signaalia, jonka arvo (esim. jännite tai virta) voi olla mahdollisten rajojen sisällä mikä tahansa. Kts. digitaalinen.

ASCII

Tietokoneissa usein käytettävä kooditaulukko, jolla kirjoitettavat merkit (aakkoset, numerot, välimerkit jne.) rinnastetaan tietokoneen käsittelemiin numeroihin. Alkuperäinen ASCII-merkistö sisältää 128 merkkiä, joista osa on varattu ns. ohjauskoodeiksi (esimerkiksi rivinvaihto, sarkain ja merkin pyyhintä).

BCD

Binäärikoodattu desimaali (engl. Binary Coded Decimal). Digitaalielektroniikassa usein käytettävä kymmenjärjestelmän lukujen esitystapa. BCD-koodissa kutakin kymmenjärjestelmän luvun yhtä numeroa vastaa neljä bittiä. Esi-

merkiksi luku 15 on BCD-koodilla esitettynä 00010101. (4 vasemmanpuoleisinta bittiä muodostavat ykkösen ja neljä seuraavaa viitosen.)

Centronics

Mikrotietokoneissa yleinen tietokoneen ja kirjoittimen välinen rinnakkaismuotoinen tiedonsiirtoliityntä. Kts. rinnakkaisliityntä.

CMOS

(engl. Complementary Metal Oxide Semiconductor) Nykyään yleinen mikropiirien valmistustekniikka, joka perustuu vähän tehoa kuluttaviin kanavatransistoreihin. Digitaalipiireissä usein CMOS-piirien kanssa rinnakkain käytetään TTL-tekniikkaa. Kts. 4000-sarja, kanavatransistori, TTL.

D-kiikku

Kiikku, joka liipaistaessa siirtää otossa (D) olevan signaalin arvon antoon (Q). Kts. kiikku.

Dataväylä

Tietokoneen väylä, joka siirtää tietoa keskusyksiköltä muistiin ja oheispiireille ja päinvastoin. Kts. väylä, osoiteväylä.

Digitaalinen

Käytetään kuvaamaan sähköistä signaalia, jonka arvo (esimerkiksi jännite tai virta) voi olla vain jokin ennalta sovituista. Yleensä digitaalisignaaleilla on vain kaksi mahdollista arvoa, nolla tai ykkönen, joita vastaavat tietyt jännitteet, esimerkiksi 0-0.5 V ja 2.4-5 V. Kts. analoginen.

Diodi

Puolijohdekomponentti, joka päästää virran läpi vain yhteen suuntaan. Diodista on myös muita erikoisempia toimintoja sisältäviä muunnoksia. Kts. kapasitanssidiodi, zenerdiodi, valodiodi.

Dynaaminen RAM-muisti (DRAM)

RAM-muisti, jonka sisältö säilyy vain rajallisen ajan luku- tai kirjoitusoperaation jälkeen. Muistin sisällön varmistamiseksi DRAM-muistia pitää virkistää määräajoin tapahtuvilla luku- tai kirjoitusoperaatioilla. Kts. staattinen RAM-muisti, RAM-muisti.

DRAM

Kts. Dynaaminen RAM-muisti

DTL

Diodi-transistori-logiikka (engl. Diode Transistor Logic). Nykyään jo käytöstä pois jäänyt loogisten peruspiirien toteutustapa.

ECL

Emitterikytketty logiikka (engl. Emitter Coupled Logic). Digitaalimikropiirien valmistustekniikka, jolla saadaan valmistettua erittäin nopeita piirejä.

Elektrolyyttikondensaattori

Kondensaattori, jonka kapasitanssi saadaan aikaan nestemäisen johteen (elektrolyytti) ja metallisen (yleensä alumiinia) kuoren rajapintaan muodostuvalla erittäin ohuella eristävällä kerroksella. Elektrolyyttikondesaattori toimii vain kun se on kytketty oikein päin.

Estovyöhyke

Puolijohdekomponenttien toiminnassa oleellinen puolijohdemateriaaliin muodostuva sähkön kulun estävä alue.

Faradi (F)

Kapasitanssin yksikkö. Määritelty kapasitanssiksi, johon tuotu yhden coulombin sähkövarauksen lisäys nostaa jännitettä yhdellä voltilla.

FET

Kts. kanavatransistori.

Flip-flop

Kts. kiikku.

Henry (H)

Induktanssin yksikkö. Määritelty induktanssiksi, jossa yhden ampeerin virran kasvu aiheuttaa magneettivuon yhden weberin kasvun.

Hertsi (Hz)

Taajuuden yksikkö. Määritelty taajuudeksi, jolla yhden sekunnin aikana ehtii tapahtua värähtelevän signaalin yksi täydellinen jakso.

hFE

Kts. tasavirtavahvistukerroin.

Induktanssi

Sähköinen suure. Komponentin induktanssi kuvaa sitä, miten voimakkaan magnettikentän sen läpi kulkeva vakiosuuruinen sähkövirta saa aikaan. Induktanssin symboli on L ja yksikkö henry (H).

Invertteri

Digitaalinen peruspiiri, joka kääntää loogisen tilan päinvastaiseksi.

Ioni

Atomi, molekyyli tai näiden ryhmä, joka on varautunut sähköisesti menettäessään tai vastaanottaessaan elektroneja.

IRQ

Kts. keskeytys.

JK-kiikku

Kiikku, jonka liipaisunjälkeistä tilaa (Qn+1) säädetään kahden ohjaussignaalin (J ja K) liipaisuhetkisillä arvoilla seuraavasti:

J	K	Qn+1
0	0	Qn
1	0	1
0	1	0
1	1	–Qn (Qn:n käänteisarvo)

Jännite

Sähköinen suure. Jännite kuvaa kahden pisteen välistä sähköisen potentiaalin eroa. Jännitteen symboli on U ja yksikkö voltti (V).

Kanavatransistori

Kanavatransistori (engl. FET = Field Effect Transistor). Transistorityyppi, jossa virran kulkua ohuen kanavan läpi säädetään kanavan ympärille muodostettavan sähkökentän avulla.

Kapasitanssi

Sähköinen suure. Kapasitanssi kuvaa komponentin kykyä varastoida sähkövarausta. Kapasitanssin symboli on C ja yksikkö faradi (F).

Kapasitanssidiodi

Diodi, jonka puolijohdekerrosten välistä kapasitanssia säädetään sähköisesti. Kapasitanssidiodeja käytetään esimerkiksi radiolaitteiden virityspiirien ominaistaajuuksien säätöön. Kts. diodi.

Kela

Komponentti, jolla muodostetaan induktanssia käärimällä johdinta useiksi samankeskisiksi renkaiksi. Kts. induktanssi.

Kellosignaali

Jaksottain toistuva signaali, jota käytetään digitaalisissa kytkennöissä toimintojen tahdistamiseen. Kellosignaalia käytetään myös monissa mittalaitteissa laskemaan tapahtumiin kulunutta aikaa.

Keskeytys

Signaali, jolla tietokoneissa oheislaitteet voivat tilapäisesti keskeyttää ohjelman suorituksen esimerkiksi jotain kiireellistä toimenpidettä varten. Keskeytyssignaaleja on koneesta riippuen yleensä useita eriarvoisia. IRQ ja NMI ovat yleisiä keskeytyssignaalien tunnuksia.

Kiikku

(engl. flip-flop) Digitaalinen komponentti, jonka antosignaali vaihdetaan ns. liipaisemalla. Liipaisu tarkoittaa yleensä erillisen kellosignaalin arvon tietynlaista muutosta, esimerkiksi muutosta nollatilasta ykköseen. Liipaisun jälkeen kiikun antosignaali muuttuu liipaisuhetken muiden ottosignaalien arvojen mukaan. Kts. D-kiikku, JK-kiikku, RS-kiikku.

Kirchhoffin lait

Kirchhoffin kaksi lakia ovat sähkötekniikan perussääntöjä ja kuuluvat:

1) Pisteeseen tulevien virtojen summa on yhtä suuri kuin siitä lähtevien virtojen summa.

2) Suljetussa virtapiirissä jännitehäviöiden summa on yhtä suuri kuin piirin lähdejännitteiden summa.

Kolmitila-anto

Digitaalipiirin anto, joka voidaan normaalien nolla- ja ykköstilojensa lisäksi asettaa korkeaimpedanssiseen tilaan. Kolmitila-annoilla voidaan esimerkiksi tietokoneen väylälle kytkeä useita väylälle lähettäviä laitteita, kunhan pidetään huolta että yhden laitteen lähettäessä muut ovat korkeaimpedanssitilassa.

Kombinaatiopiiri

Digitaalinen piiri, jonka antojen tilat riippuvat vain senhetkisistä ottojen tiloista. Kts. sekvenssipiiri.

Kondensaattori

Komponentti, jolla muodostetaan sähköistä kapasitanssia. Kondensaattori muodostuu kahdesta yleensä levymäisestä johteesta, jotka on eristetty ohuella eristekerroksella. Kts. kapasitanssi, elektrolyyttikondensaattori.

Kontrolliväylä

Tietokoneen väylä, jossa siirretään keskusyksikön ja muistin sekä oheispiirien välisiä ohjaussignaaleja, kuten muiden väylien hallintaoikeussignaaleja, keskeytyspyyntöjä jne. Kts. väylä, keskeytys, dataväylä, osoiteväylä.

Korkeaimpedanssinen tila

Kts. kolmitila-anto.

Kättely

Toisilleen tietoja lähettävien laitteiden välinen lähetys- ja vastaanottopyyntöjen ja -oikeuksien liikenne, jolla laitteet kertovat toisilleen aikeistaan ja valmiuksistaan varsinaisen tiedonsiirron yhteydessä.

LED

Kts. valodiodi.

LSI

Suurimittainen integrointi (engl. Large Scale Integration). Mikropiirien monimutkaisuuden luokka. Mikropiiri kuuluu LSI-luokkaan, jos se sisältää 100-1000 loogista porttia. Kts. SSI, MSI, VLSI.

Mikropiiri

Puolijohdesirulle tiheään rakennettu kokoelma toisiinsa kytkettyjä elektronisia komponentteja. Mikropiireillä saadaan toteutettua pienessä tilassa ja pienellä virrankulutuksella monimutkaisia kytkentöjä.

MSI

Keskimittainen integrointi (engl. Medium Scale Integration). Mikropiirien monimutkaisuuden luokka. Mikropiiri kuuluu MSI-luokkaan, jos se sisältää 12-100 loogista porttia. Kts. SSI, LSI, VLSI.

Muuntaja

Komponentti, jossa kahden kelan välisen induktanssin avulla siirretään sähköenergiaa piiristä toiseen ilman sähköistä kontaktia. Kts. induktanssi, kela.

NMI

Kts. keskeytys

Ohjelmalaskuri

Tietokoneen keskusyksikön rekisteri, joka pitää yllä parhaillaan ajettavan ohjelman muistiosoitetta. Kts. rekisteri.

Ohmi

Resistanssin yksikkö. Määritelty resistanssiksi, jossa yhden ampeerin virta aiheuttaa resistanssin päiden välillä yhden voltin jännitteen nousun.

Ohmin laki

Sääntö, joka sanoo että johtimessa kulkeva virta on suoraan verrannollinen sen päiden väliseen jännitteeseen. Voidaan myös kirjoittaa kaavamuodossa:

jännite = virta x resistanssi (U=RxI)

Osoiteväylä

Tietokoneen väylä, joka täsmentää mihin tai mistä dataväylällä kulkeva tieto on menossa tai tulossa. Kts. dataväylä, kontrolliväylä, väylä.

Pino-osoitin

Tietokoneen keskusyksikön rekisteri, joka pitää yllä ns. pino-muistin osoitetta. Kts. rekisteri.

Portti

Digitaalielektroniikan rakennuspalikka, joka toteuttaa sähköisesti jonkin loogisen perustoiminnon (JA, TAI, EI, EHDOTON-TAI sekä näiden yhdistelmät).

Puolijohde

Alkuaine, jonka sähkönjohtokyky sijoittuu johteiden ja eristeiden väliin ja jonka sähköjohtokyky paranee lämpötilan noustessa, esimerkiksi pii ja germanium. Puolijohteita käytetään lukuisissa elektroniikan komponenteissa. Tällöin puolijohdemateriaaliin sekoitetaan pieniä määriä epäpuhtauksia.

Puskuripiiri

Digitaalinen piiri, jolla vahvistetaan heikkoa virtaa antavien digitaalipiirien antosignaaleja kytkettäessä niitä esimerkiksi pitkiin siirtojohtimiin tai suureen määrään muiden digitaalipiirien ottoja.

RAM-muistipiiri

Suorahakumuistipiiri (engl. Random Access Memory). Muistipiiri, jonka muistisoluja voidaan käsitellä mielivaltaisessa järjestyksessä erillisten osoitesignaalien avulla ja jossa muistisolun käsittelyyn vaadittu aika ei merkittävästi riipu käsiteltävästä osoitteesta. Puhekielessä RAM-muistilla tarkoitetaan usein tietokoneen luku-kirjoitusmuistia, jonka sisältö häviää käyttöjännitteen katkettua.

Regulaattori

Vakavointipiiri. Käytetään teholähteissä vakavoimaan teholähteen antojännite tai -virta riippumattomaksi siitä otettavasta tehosta.

Rekisteri

Prosessorin sisäinen pieni muutaman bitin kokoinen usein tiettyyn käyttöön varattu muisti. Kts. ohjelmalaskuri, pino-osoitin.

Rele

Laite, jossa sähköisellä ilmiöllä ohjataan toista itsenäistä sähköistä ilmiötä. Useimmiten releet toimivat joko sähkömagneetilla, joka ohjaa vieressään olevaa kytkintä auki ja kiinni tai puolijohdekomponenteilla, esimerkiksi valodiodin ja valoherkän transistorin yhdistelmällä.

Resistanssi

Aineen ominaisuus vastustaa sähkövirran kulkua ja muuttaa sähkötehoa lämmöksi. Resistanssin symboli on R ja yksikkö ohmi.

Resistori

Komponentti, jolla muodostetaan virtapiirin resistanssia.

Rinnakkaisliityntä

Digitaalilaitteiden välinen kytkentä, jossa tietoa siirretään useita bittejä rinnakkain kukin omaa kanavaansa pitkin. Kts. Centronics.

RS-232

Tietokoneissa yleisesti käytettävä keskusyksikön ja oheislaitteiden välinen sarjamuotoinen tiedonsiirtoliityntä. esimerkiksi modeemit ja päätteet kytketään keskuskoneeseen usein RS-232 -liitännällä. Kts. sarjaliityntä.

RS-kiikku

Kiikku, jonka liipaisunjälkeistä tilaa (Qn+1) säädetään kahden ohjaussignaalin (R ja S) liipaisuhetkisillä arvoilla seuraavasti:

S	R	Qn+1
0	0	Qn
1	0	1
0	1	0
1	1	määrittelemätön

RTL

Vastus-transistori-logiikka (engl. Resistor-Transistor-Logic). Nykyään jo käytöstä poistunut logiikkapiirien toteutustapa.

Sarjaliityntä

Digitaalilaitteiden välinen kytkentä, jossa tietoa siirretään yhtä kanavaa pitkin yksi bitti kerrallaan. Kts. RS-232.

Seitsensegmenttinäyttö

Numeroiden selväkieliseen esittämiseen tarkoitettu seitsemästä pitkänomaisesta segmentistä koostuva näyttöelementti, jossa segmenttejä näyttämällä ja piilottamalla voidaan esittää kulmikkaannäköisinä numerot 0–9 sekä rajoitettu määrä aakkosia. Käytetään yleisesti taskulaskimissa, digitaalikelloissa ja muissa halvoissa digitaalilaitteissa.

Sekvenssipiiri

Digitaalinen piiri, jonka antojen tilat riippuvat vain paitsi senhetkisistä ottojen tiloista myös piirin ns. sisäisestä tilasta. Kts. kombinaatiopiiri.

SRAM

Kts. staattinen RAM-muisti.

SSI

Pienimittainen integrointi (engl. Small Scale Integration). Mikropiirien monimutkaisuuden luokka. Digitaalinen mikropiiri kuuluu SSI-luokkaan, jos se sisältää alle 12 loogista porttia. Kts. MSI, LSI, VLSI.

Staattinen RAM-muisti (SRAM)

RAM-muisti, jonka sisältö pysyy muuttumattomana riippumatta edellisestä luku- tai kirjoitusoperaatiosta kuluneesta ajasta. Kts. dynaaminen RAM-muisti.

Sähkövirta

Varauksenkuljettajien liike johtimessa. Varauksenkuljettajina toimivat elektronit tai ionit.

Taajuus

Suure, joka kuvaa kuinka monta täydellistä jaksoa värähtelevä signaali suorittaa aikayksikössä. Taajuuden symboli on f ja yksikkö hertsi (Hz).

Tasavirta

Sähkövirta, jossa varauksenkuljettajat liikkuvat jatkuvasti samaan suuntaan.

Tasavirtavahvistuskerroin (hFE)

Transistorin ominaisluku, joka kertoo kuinka moninkertaisen muutoksen kollektorivirrassa tietty kantatasavirran muutos aiheuttaa. Tavallisesti transistorin tasavirtavahvistuskerroin on muutamasta kymmenestä useaan sataan.

Teho

Suure, joka kuvaa energian määrän kulutusta, siirtymistä tai tuottamista. Tehon symboli on P ja yksikkö watti (W).

Toleranssi

Teknisen ominaisarvon sallittu vaihteluväli. Ilmaistaan yleensä prosentteina nimellisarvosta.

Totuustaulu

Taulukko, jolla kuvataan loogisen kytkennän antojen riippuvuutta sen otoista.

Transistori

Komponentti, jonka läpi kulkevan virran suuruutta säädetään erillisen ohjausjännitteen tai -virran avulla. Kts. kanavatransistori.

TTL

Transistori-transistori-logiikka (engl. Transistor-Transistor-Logic). Nykyään yleinen digitaalipiirien toteutustapa. TTL:n kanssa usein rinnakkain käytetään CMOS-tekniikkaan perustuvia piirejä. Kts. 74-sarja, CMOS.

Vahvistin

Sähköinen kytkentä, jonka tehtävänä on voimistaa heikon signaalin jännitettä tai virtaa.

Vaihtovirta

Sähkövirta, jossa varauksenkuljettajat kulkevat jaksottain eri suuntiin. Kts. taajuus.

Vakavointipiiri

Kts. regulaattori.

Valodiodi (LED)

(Engl. Light Emitting Diode) Diodi, joka muuttaa osan lävitseen kulkevasta sähköenergiasta valoksi. Valodiodin lähettämä valo on tarkasti yksiväristä, useimmiten punaista, vihreää, keltaista tai infrapunaa.

Vastus

Kts. resistanssi, resistori.

Virkistys

Kts. dynaaminen RAM-muisti

VLSI

Erittäin suurimittainen integrointi (engl. Very Large Scale Integration). Mikropiirien monimutkaisuuden luokka. Mikropiiri kuuluu VLSI-luokkaan, jos se sisältää yli 1000 loogista porttia. Kts. SSI, MSI, LSI.

Väylä

Kokoelma samaan käyttötarkoitukseen käytettäviä signaaleja kuljettavia johtimia. Väylään voidaan kytkeä useita laitteta, jotka voivat kukin joko lukea tai kirjoittaa väylää, kunhan kirjoittavia on kerrallaan vain yksi. Kts. osoiteväylä, dataväylä, kontrolliväylä.

Watti (W)

Tehon yksikkö. Määritelty tehoksi, jossa yksi joule energiaa kuluu, siirtyy tai tuotetaan yhdessä sekunnissa.

Z80

8-bittinen mikroprosessori, jota käytettiin yleisesti 80-luvun kotimikroissa. esimerkiksi Sinclair ZX-81, Sinclair Spectrum ja MSX-koneet sisältävät Z80:n

Zenerdiodi

Diodi, joka alkaa johtaa virtaa kun sen estosuuntaan kytketty jännite kasvaa tietyn rajan, ns. zenerjännitten yli. Zenerdiodeita käytetään vakavoitaessa jännitteitä haluttuun arvoon. Kts. regulaattori.

Elektroniikan suureita, yksiköitä ja symboleja

Suure	**Suureen symboli**	**Yksikkö**	**Yksikön symboli**
Virta	I	Ampeeri	A
Jännite	U	Voltti	V
Resistanssi	R	Ohmi	Ω
Kapasitanssi	C	Farady	F
Induktanssi	L	Henry	H
Taajuus	f	Hertsi	Hz

Indeksi

!

4000	53
4000-sarja	53
6502	52, 64, 67
74-sarja	51
7400	52
74HC	53
74L00	52
74LS00	52
74LS175	49
74LS181	69
74LS196	126
74LS273	130
74LS42	49
74LS645	65
74S00	52
7800-sarja	74
7900-piirisarja	75

A

Akku	68
Alkeishiukkaset	7
Alkuaine	7
Als-TTL	51
Alu	69
Ampeeri	8
Analoginen	33
And-gate	42
Arseeni	23
As-TTL	51
Ascii	35
Atomi	7
Atomiydin	7

B

Base	27
Bcd	35, 42
Bcd-koodi	42, 35, 42
Bidirectional buffers	64
Binary Coded Decimal	35
Binäärikoodattu desimaali	35
Binäärikoodi	34
Binäärikoodit	42
Binääriluku	35, 42
Bipolaaritransistori	30

C

CE	59
Centronics	61
Chip Enable	59
Chip Select	59
CMOS	52
Collector	27
CS	59, 66

D

D-kiikku	47
D-liitin	84
Dataväylä	62
Digitaalinen	33
Digitaalikytkentä	34
Digitaalipiiri	51
Digitaalitekniikka	34
Diode-transistor-logic	53
Diodi	24, 28
DTL	53

E

ECL 53
EHDOTON-TAI -portti 42
EHDOTON-TAI-EI -portti 42
EI-portti 42
Elektrolyyttikondensaattorit 20
Elektroni 7
Elektroniputki 51
Emitter 27
Emitter-coupled-logic 53
Emitteri 27
Energia 14
Eriste 7
Estovyöhyke 24
Exclusive-NOR -gate 42
Exclusive-OR -gate 42
EXNOR-gate = XNOR-gate 42

F

Faradi 13
FET 30
Flip-flop 44

G

Gallium 23
Germaniumdiodi 28

H

Happi 7
Henry 14
hFE 27
Hiilikalvovastus 16

I

Induktanssi 13, 14, 16
Inverter 42
IRQ 63
Irrallinen elektroni 7

J

JA-EI -portti 42
JA-portti 42
JK-kiikku 47, 49
Johteet 7
Jäähdytysripa 74
Jännite 8
Jännitteen vakavointi 73, 74, 76

K

Kaksisuuntaiset puskurit 64
Kanavatransistori 30
Kanta 27
Kapasitanssi 12, 13, 16
Kapasitanssidiodi 30
Kela 14, 21
Kellosignaali 47
Keraamiset kondensaattorit 20
Keskeytys 63
Kiikut 44
Kiillelevy 75
Kirchhoff 11
Kirchhoffin lait 11
Kollektori 27
Kolmitila-anto 59
Kombinaatiopiiri 44
Kondensaattori 13, 19
Kontrolliväylä 63
Kuivat kondensaattorit 20

Kupari 7
Kättely 137

L

L-TTL 51, 52
Lankavastus 16
Liipaisu 48
Logiikkaperhe 51, 53
Low power TTL 51
LS-TTL 51, 52
LSI 55
Läpilyönti 21, 28

M

Magneettikenttä 13
Massavastus 16
Master-slave 48
Metal-oxide-semiconductor 52
Metallikalvovastus 16
Metallioksidipuolijohde 52
Mikropiiri 33, 55
Mikroprosessori 62
MOS 51, 52
MOSFET 52
MSI 55
Muisti 55
Muistielementti 56
Muovikondensaattorit 20
Muuntaja 21

N

N-tyypin puolijohde 24
NAND-gate 42
Neutroni 7
Nimellisteho 17
NMI 63
NMOS 52
NOR-gate 42
NOT-gate 42
NPN-transistori 27
Näytteenottotaajuus 78

O

Ohjelmalaskuri 68
Ohmi 9, 10, 17
Ohmin laki 10
OR-gate 42
Osoiterekisteri 68
Osoiteväylä 62, 65

P

P-tyypin puolijohde 24
Paperikondensaattorit 20
Piidiodi 29
Piirasva 75
Pino-osoitin 68
PMOS 52
PNP-transistori 27
Portti 39
Potentiometrit 16
Protoni 7
Puhelin 51
Puolijohde 7, 23
Puskuripiiri 57
Puskurointi 62

R

Regulaattori 74
Rekisteri 68
Rele 51
RES 63
Resistanssi 9

Resistor-transistor-logic 53
Resistori 9
Rinnakkaisliityntä 61
RS-232 135
RS-kiikku 45
RTL 53

S

S-TTL 51, 52
Sarjaliityntäpiiri 61
Sarjaliitäntä 135
Seitsensegmenttinäyttö 43
Sekvenssipiiri 45
Shottky-TTL 51
SSI 55
Suodatin 92
Suure 14
Symmetrinen virtalähde 76
Säätökondensaattori 21
Säätövastukset 16
Sähkömagneetti 13
Sähkövaraus 13
Sähkövirta 8

T

TAI-EI -portti 42
TAI-portti 42
Tasavirta 14
Tasavirtavahvistuskerroin 27
Teho 9
Texas Instruments 51
TO-220-kotelo 74
Toleranssi 16
Totuustaulu 57
Totuustaulukko 39
Transistor-transistor-logic 51
Transistori 27, 30, 37, 51
TRI-STATE-OUTPUT 59
TTL 34, 51

V

Vahvistin 27
Vaihtovirta 14
Vakavointipiiri 74
Vastus 9, 16
Vety 7
Virtalähteet 73, 74, 76
VLSI 55
Voltti 8
Värikoodaus 17
Väylä 62

W

Watti 9

X

XOR-gate 42

Z

Z80 52
Zenerdiodi 28

PIKAOPPAALLA NOPEASTI SINUIKSI!

Haluatko käsikirjan, jossa asiat on selvitetty lyhyesti ja ytimekkäästi? Haluatko käsikirjan, josta hakemasi tieto löytyy helposti ja vaivatta? Haluatko kenties oppia vain käyttämäsi ohjelman uuden version uudet asiat? Vai haluatko käyttää uudesta ohjelmastasi vain pientä osaa, mutta sitäkin perusteellisemmin?

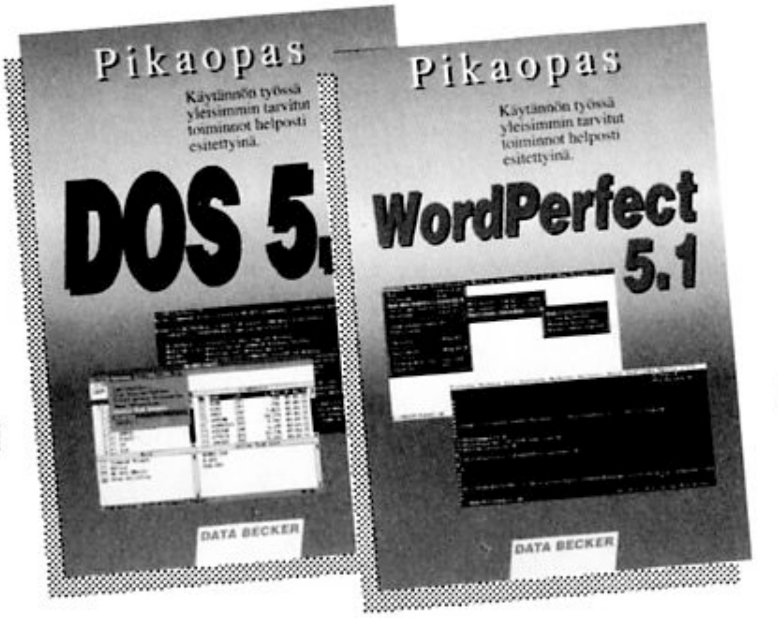
PIKAOPAS perehdyttää sinut nopeasti ja käytännönläheisesti uuteen ohjelmaasi. PIKAOPPAAN ulkoasussa on otettu huomioon käyttäjä: sivun reunassa olevien otsikoiden avulla asiat löytyvät helposti kirjaa selaamalla. Reunahuomautusten avulla erityistä huomiota vaativat kohdat on helppo huomata.

PIKAOPPAASSA on runsaasti havainnollisia kuvia. Ongelmatilanteet on käsitelty vaihe vaiheelta, näppäin näppäimeltä . Yhteenvedoissa annetaan johdonmukaiset toimintaohjeet ja kuvaillaan toiminnot selkeästi.

DOS 5.0

- asennus
- tiedostotyöskentely
- hakemistot
- DOS-Shell-ohjelmat
- komentotason työskentely
- tekstinkäsittely DOS-editorilla
- tiedon turvaaminen
- asetusten optimointi

ovh. 125,-



WORD-PERFECT 5.1

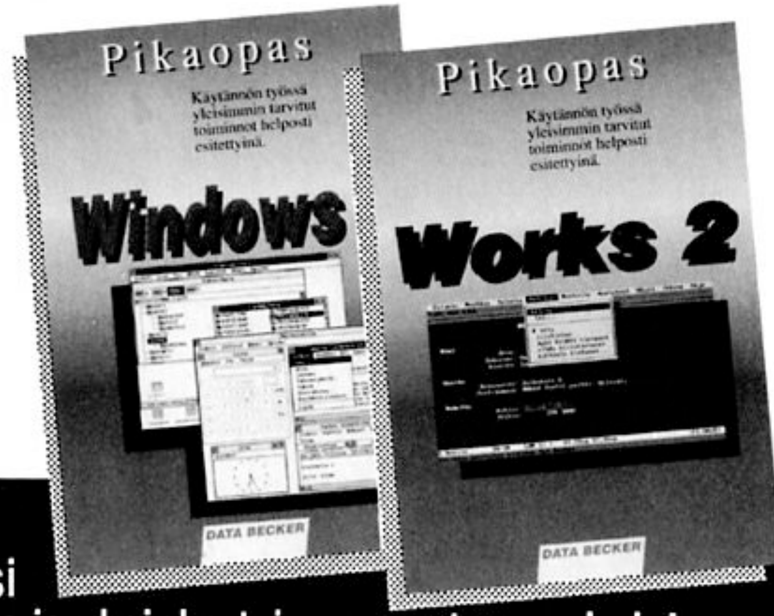
- asennus
- perustoiminnot
- kirjeet
- laskut
- ryhmäkirjeet
- julkaisut
- dokumentin muotoilu

ovh. 125,-

WINDOWS 3

- asennus
- hiiren käyttö
- ohjelmapalvelija
- tiedostopalvelija
- apuohjelmien käyttö
- vinkkejä

ovh. 125,-



WORKS 2

- asennus
- perustoiminnot
- kirjeet
- videokasettikortisto
- taulukkolaskenta
- pääteohjelma
- makrojen teko

ovh. 125,-

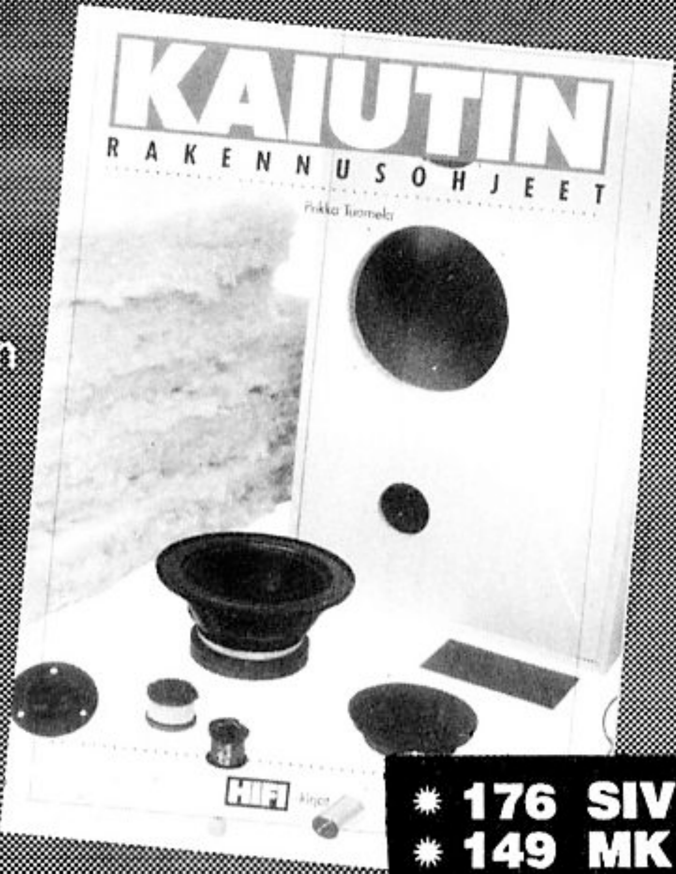
PIKAOPPAAN avulla käytät uutta ohjelmaasi heti.
Vaivattomampaa ja yksinkertaisempaa tapaa ei ole!

TILAA PUHELIMITSE 90 - 120 5911!

LAATUKAIUTTIMET EDULLISESTI

KUN RAKENNAT KAIUTTIMET ITSE, VOIT PARANTAA LAITTEISTOSI ÄÄNENLAATUA KOHTUULLISIN KUSTANNUKSIN.

HIFI-lehden julkaisema Kaiutinrakennusohjeet-kirja sisältää täydelliset ohjeet 17 erilaisesta kaiuttimesta. Kirja perustuu HIFI-lehdessä vuosina 1981 - 1989 julkaistuihin ohjeisiin. Kirjaa varten niihin on tehty muutoksia ja ne on ajanmukaistettu.



Kirjasta löydät tiedot mm. seuraavista asioista:

- Kotelomateriaalin valinta
- Kotelon rakenne
- Reunaheijastuksen välttäminen
- Refleksiputken mitoitus
- Vaimennusaineen valinta ja määrä
- Jakosuotimien osien sijoittelu
- Komponenttien laatu
- Kelojen ja kondensaattorien kytkeminen
- Mittaustulosten tulkinta

✷ 176 SIVUA
✷ 149 MK

Kaiuttimien suunnittelusta kiinnostuneelle kirjassa on liitteenä elementtien datalehdet, joista selviävät niiden sähköakustiset ominaisuudet.

TILAA NYT NUMEROSTA
90 - 120 671!

Bittinikkarin käsikirja

Bittinikkarin käsikirja on elektroniikkaharrastajan perusteos, joka soveltuu niin harrastustaan vasta aloittavalle kuin kokeneemmallekin rakentelijalle. Kirja keskittyy erityisesti digitaalielektroniikkaan ja kotimikroihin liitettävien laitteiden rakenteluun.

Kirja jakaantuu kahteen osaan. Alussa oleva teoriaosa johdattelee aloittelevan harrastajan sähkötekniikan alkeista aina mikrotietokoneen rakenteeseen. Teoriaosa toimii myös kokeneemman harrastajan pienenä lähdeteoksena.

Loppuosa kirjasta sisältää runsaasti valmiita sovelluksia. Mukana on sekä itsenäisiä laitteita että kotimikron lisälaitteita. Nämä laitteet esittävät kokemattomalle harrastajalle teoriaosan sovelluksia ja antavat kokeneemmalle ideoita ja valmiita elektroniikan rakennuspalikoita. Suurin osa sovelluksista sisältää laitteiden kytkentäkaavioiden ja toiminnan selityksen lisäksi myös piirilevymallit.

Kirjan kirjoittaja tekniikan yo Ari Paananen on harrastanut elektroniikkaa 15 vuotta. Hänen rakenteluohjeitaan on julkaistu muun muassa MikroBITTI-lehdessä vuodesta 1986 lähtien.

SANOMAPRINT
ERIKOISLEHDET
TECNOPRESS

ISBN 951-832-025-X